IJCAI·11

Proceedings of the IJCAI-2011 Workshop on Social Choice and Artificial Intelligence

Edith Elkind, Ulle Endriss and Jérôme Lang (eds.)

Barcelona, 16 July 2011

Programme Committee

Craig Boutilier	University of Toronto
Ioannis Caragiannis	University of Patras
Vincent Conitzer	Duke University
Edith Elkind (*co-chair*)	Nanyang Technological University
Ulle Endriss (*co-chair*)	University of Amsterdam
Gábor Erdélyi	Nanyang Technological University
Piotr Faliszewski	AGH University of Science and Technology
Paul Harrenstein	Technical University Munich
Wiebe van der Hoek	University of Liverpool
Sébastien Konieczny	CNRS & Université d'Artois
Sarit Kraus	Bar Ilan University & University of Maryland
Jérôme Lang (*co-chair*)	CNRS & Université Paris-Dauphine
Fangzhen Lin	Hong Kong University of Science and Technology
Ariel Procaccia	Harvard University
Jeff Rosenschein	Hebrew University of Jerusalem
Francesca Rossi	University of Padova
Moshe Tennenholtz	Technion & Microsoft Isreal R&D Center
Toby Walsh	NICTA & University of New South Wales

Additional Reviewers

Stéphane Airiau	Ning Ding
Umberto Grandi	Omer Lev
Reshef Meir	Nina Narodytska
Maria Silvia Pini	Daniele Porello
Zinovi Rabinovich	Michael Zuckerman

Workshop Website

`http://www.illc.uva.nl/COMSOC/IJCAI-2011/`

Preface

Social choice theory is the study of mechanisms for collective decision making, such as election systems or protocols for fair division. Computational social choice addresses problems at the interface of social choice theory with computer science, either by using concepts and methods from social choice theory to solve problems arising in computer science (such as webpage ranking), or by using techniques from computer science to solve (or reformulate) problems of social choice (such as designing social choice rules that are computationally hard to manipulate).

These proceedings present the latest developments in the area of computational social choice, with a particular focus on questions at the interface of social choice and artificial intelligence. They contain 19 papers that will be presented at the IJCAI Workshop on Social Choice and Artificial Intelligence, which is part of the workshop programme of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011), to be held in Barcelona in July 2011. The papers have been reviewed in depth by the members of the programme committee and several external reviewers. In line with the open and informal nature of the workshop, we have selected both technically mature contributions to the field and papers reporting on work in progress. Together they cover a wide range of topics, including the design and analysis of voting rules, non-truthful behaviour in elections, coalition formation, tournaments, fair division, stable marriage problems, and judgement aggregation. The copyright of the papers in this volume remains with the individual authors.

We would like to thank all authors for their submissions and the PC members and reviewers for their help in selecting the contributions to be presented at the workshop.

Singapore, Amsterdam & Paris — E.E., U.E. & J.L.
May 2011

Table of Contents

Pareto Optimality in Coalition Formation

Haris Aziz Felix Brandt Paul Harrenstein
Department of Informatics
Technische Universität München
85748 Garching bei München, Germany
{aziz,brandtf,harrenst}@in.tum.de

Abstract

A minimal requirement on allocative efficiency in the social sciences is Pareto optimality. In this paper, we exploit a strong structural connection between Pareto optimal and perfect partitions that has various algorithmic consequences for coalition formation. In particular, we show that computing and verifying Pareto optimal partitions in general hedonic games and B-hedonic games is intractable while both problems are tractable for roommate games and W-hedonic games. The latter two positive results are obtained by reductions to maximum weight matching and clique packing, respectively.

1 Introduction

Topics concerning coalitions and coalition formation have come under increasing scrutiny of computer scientists. The reason for this may be obvious. For the proper operation of distributed and multiagent systems, cooperation may be required. At the same time, collaboration in very large groups may also lead to unnecessary overhead, which may even exceed the positive effects of cooperation. To model such situations formally, concepts from the social and economic sciences have proved to be very helpful and thus provide the mathematical basis for a better understanding of the issues involved.

Coalition formation games, which were first formalized by Drèze and Greenberg [1980], model coalition formation in settings in which utility is *non-transferable*. In many such situations it is natural to assume that a player's appreciation of a coalition structure only depends on the coalition he is a member of and not on how the remaining players are grouped. Initiated by Banerjee *et al.* [2001] and Bogomolnaia and Jackson [2002], much of the work on coalition formation now concentrates on these so-called *hedonic games*. In this paper, we focus on Pareto optimality and individual rationality in this rich class of coalition formation games.

The main question in coalition formation games is which coalitions one may reasonably expect to form. To get a proper formal grasp of this issue, a number of stability concepts have been proposed for hedonic games—such as the core or Nash stability—and much research concentrates on conditions for existence, the structure, and computation of stable and efficient partitions. *Pareto optimality*—which holds if no coalition structure is strictly better for some player without being strictly worse for another—and *individual rationality*—which holds if every player is satisfied in the sense that no player would rather be on his own—are commonly considered minimal requirements for any reasonable partition.

Another reason to investigate Pareto optimal partitions algorithmically is that, in contrast to other stability concepts like the core, they are guaranteed to exist. This even holds if we additionally require individual rationality. Moreover, even though the *Gale-Shapley algorithm* returns a core stable matching for marriage games, it is already NP-hard to check whether the core is empty in various classes and representations of hedonic games, such as roommate games [Ronn, 1990], general hedonic games [Ballester, 2004], and games with $\mathscr{B}$- and $\mathscr{W}$-preferences [Cechlárová and Hajduková, 2004a,b]. Interestingly, when the status-quo partition cannot be changed without the mutual consent of all players, Pareto optimality defines stability [Morrill, 2010].

In this paper, we investigate both the problem of finding a Pareto optimal and individually rational partition and the problem of deciding whether a partition is Pareto optimal. In particular, our results concern *general hedonic games*, *B-hedonic* and *W-hedonic games* (two classes of games in which each player's preferences over coalitions are based on his most preferred and least preferred player in his coalition, respectively), and *roommate* games.

Many of our results, both positive and negative, rely on the concept of *perfection* and how it relates to Pareto optimality. A *perfect* partition is one that is most desirable for every player. We find (*a*) that under extremely mild conditions, NP-hardness of finding a perfect partition implies NP-hardness of finding a Pareto optimal partition (Lemma 1), and (*b*) that under stronger but equally well-specified circumstances, feasibility of finding a perfect partition implies feasibility of finding a Pareto optimal partition (Lemma 2). The latter we show via a Turing reduction to the problem of computing a perfect partition. At the heart of this algorithm, which we refer to as the *Preference Refinement Algorithm (PRA)*, lies a fundamental insight of how perfection and Pareto optimality are related. It turns out that a partition is Pareto optimal for a particular preference profile if and only if the partition is perfect for another but related one (Theorem 1). In this way PRA is also

applicable to any other discrete allocation setting.

For general allocation problems, *serial dictatorship*—which chooses subsequently the most preferred allocation for a player given a fixed ranking of all players—is well-established as a procedure for finding Pareto optimal solutions [see, e.g., Abdulkadiroğlu and Sönmez, 1998]. However, it is only guaranteed to do so, if the players' preferences over outcomes are strict, which is not feasible in many compact representations. Moreover, when applied to coalition formation games, there may be Pareto optimal partitions that serial dictatorship is unable to find, which may have serious repercussions if also other considerations, like fairness, are taken into account. By contrast, PRA handles weak preferences well and and is complete in the sense that it may return any Pareto optimal partition, provided that the subroutine that calculates perfect partitions can compute any perfect partition (Theorem 2).

2 Preliminaries

In this section, we review the terminology and notation used in this paper.

Hedonic games Let N be a set of n players. A *coalition* is any non-empty subset of N. By $\mathcal{N}_i$ we denote the set of coalitions player i belongs to, i.e., $\mathcal{N}_i = \{S \subseteq N : i \in S\}$. A *coalition structure*, or simply a *partition*, is a partition π of the players N into coalitions, where $\pi(i)$ is the coalition player i belongs to.

A *hedonic game* is a pair (N, R), where $R = (R_1, \ldots, R_n)$ is a *preference profile* specifying the preferences of each player i as a binary, complete, reflexive, and transitive *preference relation* R_i over $\mathcal{N}_i$. If R_i is also anti-symmetric we say that i's preferences are *strict*. We adopt the conventions of social choice theory by writing $S\ P_i\ T$ if $S\ R_i\ T$ but not $T\ R_i\ S$—i.e., if i *strictly prefers* S to T—and $S\ I_i\ T$ if both $S\ R_i\ T$ and $T\ R_i\ S$—i.e., if i is *indifferent* between S and T.

For a player i, a coalition S in $\mathcal{N}_i$ is *acceptable* if for i being in S is at least preferable as being alone—i.e., if $S\ R_i\ \{i\}$—and *unacceptable* otherwise.

In a similar fashion, for X a subset of $\mathcal{N}_i$, a coalition S in X is said to be *most preferred in X by i* if $S\ R_i\ T$ for all T in X and *least preferred in X by i* if $T\ R_i\ S$ for all $T \in X$. In case $X = \mathcal{N}_i$ we generally omit the reference to X. The sets of most and least preferred coalitions in X by i, we denote by $\max_{R_i}(X)$ and $\min_{R_i}(X)$, respectively.

In hedonic games players are only interested in the coalition they are in. Accordingly, preferences over coalitions naturally extend to preferences over partitions and we write $\pi\, R_i\, \pi'$ if $\pi(i)\, R_i\, \pi'(i)$. We also say that partition π is *acceptable* or *unacceptable* to a player i according to whether $\pi(i)$ is acceptable or unacceptable to i, respectively. Moreover, π is *individually rational* if π is acceptable to all players. A partition π is *Pareto optimal in R* if there is no partition π' with $\pi'\, R_j\, \pi$ for all players j and $\pi'\, P_i\, \pi$ for at least one player i. Partition π is, moreover, said to be *weakly Pareto optimal in R_i* if there is no π' with $\pi'\, P_i\, \pi$ for all players i.

Classes of hedonic games The number of potential coalitions grows exponentially in the number of players. In this sense, hedonic games are relatively large objects and for algorithmic purposes it is often useful to look at classes of games that allow for concise representations.

For *general hedonic games*, we will assume that each player expresses his preferences only over his acceptable coalitions. This representation is alternatively known as *Representation by Individually Rational Lists of Coalitions* [Ballester, 2004].

We now describe classes of hedonic games in which the players' preferences over coalitions are induced by their preferences over the other players. For R_i such preferences of player i over players, we say that a player j is *acceptable* to i if $j\, R_i\, i$ and *unacceptable* otherwise. Any coalition containing an unacceptable player is unacceptable to player i.

Roommate games. The class of *roommate games*, which are well-known from the literature on matching theory, can be defined as those hedonic games in which only coalitions of size one or two are acceptable.

B-hedonic and W-hedonic games. For a subset J of players, we denote by $\max_{R_i}(J)$ and $\min_{R_i}(J)$ the sets of the most and least preferred players in J by i, respectively. We will assume that $\max_{R_i}(\emptyset) = \min_{R_i}(\emptyset) = \{i\}$. In a *B-hedonic game* the preferences R_i of a player i over players extend to preferences over coalitions in such a way that, for all coalitions S and T in $\mathcal{N}_i$, we have $S\ R_i\ T$ if and only if $\max_{R_i}(S \setminus \{i\})\, R_i\, \max_{R_i}(T \setminus \{i\})$ or some j in T is unacceptable to i. Analogously, in a *W-hedonic game* (N, R), we have $S\ R_i\ T$ if and only if $\min_{R_i}(S \setminus \{i\})\, R_i\, \min_{R_i}(T \setminus \{i\})$ or some j in T is unacceptable to i.[1]

3 Perfection and Pareto Optimality

Pareto optimality constitutes a rather minimal efficiency requirement on partitions. A much stronger property is that of *perfection*. We say that a partition π is *perfect* if $\pi(i)$ is a most preferred coalition for all players i. Thus, every perfect partition is Pareto optimal but not necessarily the other way round. Perfect partitions are obviously very desirable, but, in contrast to Pareto optimal ones, they are not guaranteed to exist. Still, a strong structural connection exists between the two concepts, which, in the next section, we exploit in our algorithm for finding Pareto optimal partitions.

The problem of finding a perfect partition (PP) we formally specify as follows: given a preference profile R, find a perfect partition for R and if no perfect partition exists in R, output "none".

We will later see that the complexity of PP depends on the specific class of hedonic games that is being considered. By contrast, the related problem of *checking* whether a partition is perfect is an almost trivial problem for virtually all reasonable classes of games. If perfect partitions exist, they clearly coincide with the Pareto optimal ones. Hence, an oracle to compute a Pareto optimal partition can be used to solve PP.

[1] W-hedonic games are equivalent to hedonic games with $\mathscr{W}$-preferences if individually rational outcomes are assumed. Unlike hedonic games with $\mathscr{B}$-preferences, B-hedonic games are defined in analogy to W-hedonic games and the preferences are not based on coalition sizes [cf. Cechlárová and Hajduková, 2004a].

If this Pareto optimal partition is perfect we are done, if it is not, no perfect partitions exist. Thus, we obtain the following lemma, which we will invoke in our hardness proofs for computing Pareto optimal partitions.

Lemma 1 *For every class of hedonic games for which checking whether a given partition is perfect can be solved in polynomial time, NP-hardness of* PP *implies NP-hardness of computing a Pareto optimal partition.*

It might be less obvious that a procedure solving PP can also be deployed as an oracle for an algorithm to compute Pareto optimal partitions. To do so, we first give a characterization of Pareto optimal partitions in terms of perfect partitions, which forms the mathematical heart of the Preference Refinement Algorithm to be presented in the next section.

This characterization depends on the concept of a coarsening of a preference profile and the lattices these coarsenings define. To make things precise, we say that a preference profile $R = (R_1, \ldots, R_n)$ is a *coarsening of* or *coarsens* another preference profile $R' = (R'_1, \ldots, R'_n)$ whenever for every player i we have $R'_i \subseteq R_i$. In that case we also say that R' *refines* R and write $R \leq R'$. Moreover, we write $R < R'$ if $R \leq R'$ but not $R' \leq R$. Thus, if R' refines R, i.e., if $R \leq R'$, then for each i and all coalitions S and T we have that $S\, R'_i\, T$ implies $S\, R_i\, T$, but not necessarily the other way round. Intuitively, a player i may be indifferent in R between coalitions over which i entertains strict preferences in R'. It is worth observing that, if a partition is perfect in some preference profile R, then it is also perfect in any coarsening of R. The same holds for Pareto optimal partitions.

For preference profiles R and R' with $R \leq R'$, let $[R, R']$ denote the set $\{R'' : R \leq R'' \leq R'\}$, i.e., $[R, R']$ is the set of all coarsenings of R' that are not coarser than R. Then, $([R, R'], \leq)$ is a complete lattice with R and R' as bottom and top element, respectively. We say that R *covers* R' if R is a minimal refinement of R', i.e., if $R' < R$ and there is no R'' such that $R' < R'' < R$. R *strongly covers* R' if among all preference profiles that cover R', R is one that, for all players, allows for a maximal number of most preferred alternatives, i.e., $\max_{R''_i}(\mathcal{N}_i) \subseteq \max_{R_i}(\mathcal{N}_i)$ for all players i and each R'' that covers R'. We are now in a position to prove the following theorem, which characterizes Pareto optimal partitions given a preference profile R as those that are perfect in particular coarsenings R' of R. These R' are such that no perfect partitions exist in any preference profile that strongly covers R'.

Theorem 1 *Let $(N, R^\top)$ and $(N, R^\bot)$ be hedonic games and π a partition such that $R^\bot \leq R^\top$ and π is a perfect partition in $R^\bot$. Then, π is Pareto optimal in $R^\top$ if and only if there is some $R \in [R^\bot, R^\top]$ such that (i) π is a perfect partition in R and (ii) there is no perfect partition for any $R' \in [R^\bot, R^\top]$ that strongly covers R.*

Proof: For the if-direction, assume there is some $R \in [R^\bot, R^\top]$ such that π is perfect in R and there is no perfect partition in any $R' \in [R^\bot, R^\top]$ that strongly covers R. (Observe that this implies that, for all i, R_i and $R_i^\top$ coincide on coalitions less preferred by i than $\pi(i)$.) For contradiction, also assume π is not Pareto optimal in $R^\top$. Then, there is some π' such that $\pi'\, R_j^\top\, \pi$ for all j and $\pi'\, P_i^\top\, \pi$ for some i. By $R \leq R^\top$ and transitivity of preferences, π' is a perfect partition in R as well. Let π'' be such that $\pi''(i) \in \min_{R_i^\top}(\max_{R_i}(\mathcal{N}_i))$ and define $R'_i = R_i \setminus \{(X, Y) : \pi''(i)\, R_i^\top\, X \text{ and } Y\, P_i^\top\, \pi''(i)\}$. Thus, $\pi''(i)$ is one of i's least preferred coalitions according to $R_i^\top$ among i's most preferred coalitions in R_i. Intuitively, R'_i is exactly like R_i be it that i strictly prefers Y to X in R'_i if $X \in \min_{R_i^\top}(\max_{R_i}(\mathcal{N}_i))$ and $Y\, P_i^\top\, X$. Observe that $R' = (R_1, \ldots, R_{i-1}, R'_i, R_{i+1}, \ldots, R_n)$ is in $[R^\bot, R^\top]$ and covers R. By choice of π'', R' even strongly covers R. Moreover, as $\pi'\, P_i^\top\, \pi$ and, therefore, $\pi' \notin \min_{R_i^\top}(\max_{R_i}(\mathcal{N}_i))$, π' is still a perfect partition in R', a contradiction.

For the only-if direction assume that π is Pareto optimal in $R^\top$. Let R be the finest coarsening of $R^\top$ in which π is perfect. Observe that $R = (R_1, \ldots, R_n)$ can be defined such that $R_i = R_i^\top \cup \{(X, Y) : X\, R_i^\top\, \pi \text{ and } Y\, R_i^\top\, \pi\}$ for all i. Also observe that $R^\bot \leq R$. If $R = R^\top$, we are done immediately. Otherwise, consider an arbitrary $R' \in [R^\bot, R^\top]$ that strongly covers R and assume for contradiction that there is some perfect partition π' in R'. Then, in particular, $\pi'\, R'_k\, \pi$ for all k. Since R' covers R, there is exactly one i with $R'_i \neq R_i$, whereas $R'_j = R_j$ for all $j \neq i$. As π is perfect in R, we also have $\pi\, R'_j\, \pi'$ for all $j \neq i$. With R' being a finer coarsening of $R^\top$ than R, however, π is not perfect in R'. Hence, it is not the case that $\pi\, R'_i\, \pi'$ and, therefore, $\pi'\, P'_i\, \pi$. We may now conclude that π is not Pareto optimal in R'. Since, $R' \leq R^\top$, moreover, π not Pareto optimal in $R^\top$ either, a contradiction. □

4 The Preference Refinement Algorithm

In this section, we present the *Preference Refinement Algorithm (PRA)*, a general algorithm to compute Pareto optimal and individually rational partitions. The algorithm invokes an oracle solving PP and is based on the formal connection between Pareto optimality and perfection made explicit in Theorem 1.

The idea underlying the algorithm is as follows. To calculate a Pareto optimal and individually rational partition for a hedonic game (N, R), first find that coarsening R' of R in which each player is indifferent among all his acceptable coalitions and his preferences among unacceptable coalitions are as in R. In this coarsening, a perfect and individually rational partition—which we also refer to as the *coarsest acceptable coarsening*—is guaranteed to exist. From there on, start moving up in the lattice $([R', R], \leq)$ to strongly covering preference profiles for which a perfect partition exists, until you reach a preference profile for which this is no longer possible. By calculating a perfect partition for this last preference profile, in virtue of Theorem 1, you find a Pareto optimal partition for R. A formal specification of PRA is given in Algorithm 1. It is worth mentioning that Algorithm 1 is an anytime algorithm that can return an intermediate result when stopped prematurely.

Theorem 2 *For any hedonic game (N, R),*

(*i*) *PRA returns an individually rational and Pareto optimal partition.*

Algorithm 1 Preference Refinement Algorithm (PRA)

Input: Hedonic game (N, R)
Output: Pareto optimal and individually rational partition

1 $Q_i \leftarrow R_i \cup \{(X, Y) : X\, R_i\, \{i\}$ and $Y\, R_i\, \{i\}\}$, for each $i \in N$
2 $Q \leftarrow (Q_1, \ldots, Q_n)$
3 $J \leftarrow N$
4 **while** $J \neq \emptyset$ **do**
5 $\quad i \in J$
6 $\quad S \in \min_{R_i}(\max_{Q_i}(\mathcal{N}_i))$
7 $\quad Q_i' \leftarrow Q_i \setminus \{(X, Y) : S\, R_i\, X$ and $Y\, P_i\, S\}$
8 $\quad Q' \leftarrow (Q_1, \ldots, Q_{i-1}, Q_i', Q_{i+1}, \ldots, Q_n)$
9 $\quad$ **if** PP$(N, Q') \neq$ none **then**
10 $\quad\quad Q \leftarrow Q'$
11 $\quad$ **else**
12 $\quad\quad J \leftarrow J \setminus \{i\}$
13 $\quad$ **end if**
14 **end while**
15 **return** PP(N, Q)

(*ii*) *For every individually rational and Pareto optimal partition π', there is an execution of PRA that returns a partition π such that $\pi\, I_i\, \pi'$ for all i in N.*

Proof: For (*i*), we prove that during the running of PRA, for each assignment of Q, there exists a perfect partition π for that assignment. This claim certainly holds for the first assignment of Q which is the coarsest acceptable coarsening of R. Furthermore, Q is only refined via the strong covering relation (Steps 6 through 7), if there exists a perfect partition for a strong covering of Q. Let Q^* be the final assignment of Q. Then, we argue that the partition π returned by PRA is Pareto optimal and individually rational. By Theorem 1, if π were not Pareto optimal, there would exist a strong covering of Q^* for which a perfect partition still exists and Q^* would not be the final assignment of Q. Since, each player at least gets one of his acceptable coalitions, π is also individually rational.

For (*ii*), first observe that, by Theorem 1, for each Pareto optimal and individually rational partition π for a preference profile R there is some coarsening R^* of R where π is perfect and no perfect partitions exist for any strong covering of R^*. By individual rationality of π, it follows that R^* is a refinement of the initial assignment of Q. An appropriate number of strong coverings of the initial assignment of Q with respect to each player results in a final assignment Q^* of Q to R^*. The perfect partition for Q^* that is returned by PRA is then such that $\pi\, I_i\, \pi'$ for all i in N. □

Note that for each player's preferences over coalitions induces equivalence classes in which a player is indifferent between coalitions in the same equivalence class. We specify the conditions under which PRA runs in polynomial time.

Lemma 2 *Let (N, R) be a hedonic game such that for each player the number of equivalence classes of acceptable outcomes is polynomial in the input, the coarsest acceptable coarsening of R as well as the strong coverings of each of its refinements can be computed in polynomial time, and* PP *can be solved in polynomial time for all coarsenings of R. Then, PRA runs in polynomial time.*

Proof: Under the given conditions, we prove that PRA runs in polynomial time. In each iteration of the while-loop, either the preference profile Q is strongly covered (Step 10) or a player i which cannot be further improved is removed from J (Step 12). Both of these steps take polynomial time due to the conditions specified. Since each player has a polynomial number of acceptable equivalence classes in R_i, there can only be a polynomial number of reassignments of Q and therefore the while-loop iterates a polynomial number of times. As the crucial subroutine PP (Step 9) takes polynomial time, PRA runs in polynomial time. □

PRA applies not only to general hedonic games but to many natural classes of hedonic games in which equivalence classes (of possibly exponentially many coalitions) for each player are implicitly defined.[2]

Note that PRA as it is presented does not leverage the potential benefit of preferences being strict because when preferences are coarsened, the strictness of the preferences is lost and PP becomes NP-hard (see Theorem 3). *Serial dictatorship* is a well-studied mechanism in resource allocation, in which an arbitrary player is chosen as the 'dictator' who is then given his most favored allocation and the process is repeated until all players or resources have been dealt with. In the context of coalition formation, *serial dictatorship* is well-defined only if in every iteration, the dictator has a unique most preferred coalition.

Proposition 1 *For general hedonic games, W-hedonic games, and roommate games, a Pareto optimal partition can be computed in polynomial time when preferences are strict.*

Proposition 1 follows from the application of serial dictatorship to hedonic games with strict preferences over the coalitions. If the preferences over coalitions are not strict, then the decision to assign one of the favorite coalitions to the dictator may be sub-optimal. Serial dictatorship does not work for hedonic games in which preferences over coalitions are not strict, not even for B-hedonic games with strict preferences over players. Observe that PRA can be tweaked so as to obtain an individually rational version of the serial dictatorship algorithm, which also achieves the positive results of Proposition 1. Abdulkadiroğlu and Sönmez [1998] showed that in the case of strict preferences and house allocation settings, every Pareto optimal allocation can be achieved by serial dictatorship. In the case of coalition formation, however, it is easy to construct a four-player hedonic game with strict preferences for which there is a Pareto optimal partition that serial dictatorship cannot return.

5 Computational results

In this section, we consider the problem of VERIFICATION (verifying whether a given partition is Pareto optimal) and COMPUTATION (computing a Pareto optimal partition).

[2] For example, in W-hedonic games, $\max_{R_i}(N)$ specifies the set of favorite players of player i but can also implicitly represent all those coalitions S such that the least preferred player in S is also a favorite player for i.

5.1 General hedonic games

As shown in Proposition 1, Pareto optimal partitions can be found efficiently for general hedonic games with strict preferences. If preferences are not strict, the problem becomes NP-hard. We can prove the following statement by utilizing Lemma 1 and showing that PP is NP-hard by a reduction from ExactCoverBy3Sets (X3C).

Theorem 3 *For a general hedonic game, computing a Pareto optimal partition is NP-hard even when each player has a maximum of four acceptable coalitions and the maximum size of each coalition is three.*

Interestingly, *verifying* Pareto optimality is coNP-complete even for strict preferences.

Theorem 4 *For any general hedonic game, verifying whether a partition π is Pareto optimal and whether π is weakly Pareto optimal is coNP-complete even when preferences are strict and π consists of the grand coalition of all players.*

5.2 Roommate games

For roommate games, we observe that PP is equivalent to solving a perfect matching of the graph in which two vertices (players) are connected if and only if they consider each other as a favorite player. Therefore, we obtain the following as a corollary of Lemma 2.

Theorem 5 *For roommate games, an individually rational and Pareto optimal coalition can be computed in polynomial time.*

We found that in the case of general hedonic games, verifying Pareto optimality can be significantly harder than computing a Pareto optimal partition when preferences are strict. Abraham and Manlove [2004] and Morrill [2010] showed that there are efficient algorithms to verify whether a partition is Pareto optimal for roommate games with strict preferences. The more general case of non-strict preferences is left open.[3] We answer this problem in the next theorem.

Theorem 6 *For roommate games, it can be checked in polynomial time whether a partition is Pareto optimal.*

Proof sketch: We reduce the problem to computing a maximum weight matching of a graph.

For roommate game (N, R), let π be the partition which we want to check for Pareto optimality. Since π contains coalitions of size one or two, we can construct an undirected graph $G = (V, E)$ where $V = N \cup (N \times \{0\})$, $E = V \times V \setminus (\{\{i, j\} : \pi(i)\, P_i\, \{i\}\} \cup \{\{i, (i, 0)\} : \pi(i)\, P_i\, \{i\}\}$. For graph (V, E), consider the matching $M = \{S \in \pi : |S| = 2\} \cup \{\{i, (i, 0)\} : \{i\} \in \pi\}$.

We now define a weight function such that for all $i \in V$, $w_i : E \to \mathbb{R}^+$ where w_i is defined inductively in the following way: $w_{(i,0)}(e) = 0$ for all e such that $(i, 0) \in e \in E$ and $i \in N$; $w_i(\pi(i)) = n$ if $\pi(i) \neq \{i\}$ and $\pi(i) = \{i, j\}$; $w_i(\{i, (i, 0)\}) = n$ if $\pi(i) = \{i\}$; $w_i(S) = -n$ if $i \notin S$; $w_i(T) = w_i(S) + 1/n$ if there is a coalition T such that $i \in T$, $T\, P_i\, S$, and there exists no coalition T' such that $T\, P_i\, T'\, P_i\, S$; and $w_i(T) = w_i(S)$ if $S\, R_i\, \pi(i)$ and T is coalition such that $T\, I_i\, S$.

Define a weight function $w' : E \to \mathbb{R}^+$ such that for any $S = \{i, j\} \in E$, $w'(S) = w_i(S) + w_j(S)$. For $E'' \subseteq E$, denote by $w'(E'')$, the value $\sum_{e \in E''} w'(e)$. We can then prove that π is Pareto optimal if and only if π is the maximum weight matching of $G^{w'}$, the graph G, weighted by weight function w'. The complete proof is omitted due to space limitations. Since we have a linear-time reduction to maximum weight matching [Gabow and Tarjan, 1991], the complexity of the algorithm is $O(n^3)$. □

Note that Theorem 6 allows us to find a Pareto optimal Pareto improvement for any given partition if the partition is not Pareto optimal.

5.3 W-hedonic games

We now turn to Pareto optimality in W-hedonic games.

Theorem 7 *For W-hedonic games, a partition that is both individually rational and Pareto optimal can be computed in polynomial time.*

Proof sketch: The statement follows from Lemma 2 and the fact that PP can be solved in polynomial time for W-hedonic games. The latter is proved by a polynomial-time reduction of PP to a polynomial-time solvable problem called *clique packing*.

We first introduce the more general notion of graph packing. Let $\mathscr{F}$ be a set of undirected graphs. An $\mathscr{F}$-packing of a graph G is a subgraph H such that each component of H is (isomorphic to) a member of $\mathscr{F}$. The size of $\mathscr{F}$-packing H is $|V(H)|$. We will informally say that vertex i is *matched* by $\mathscr{F}$-packing H if i is in a connected component in H. Then, a maximum $\mathscr{F}$-packing of a graph G is one that matches the maximum number of vertices. It is easy to see that computing a maximum $\{K_2\}$-packing of a graph is equivalent to maximum cardinality matching. Hell and Kirkpatrick [1984] and Cornuéjols *et al.* [1982] independently proved that there is a polynomial-time algorithm to compute a maximum $\{K_2, \ldots, K_n\}$-packing of a graph. Cornuéjols *et al.* [1982] note that finding a $\{K_2, \ldots, K_n\}$-packing can be reduced to finding a $\{K_2, K_3\}$-packing.

We are now in a position to reduce PP for W-hedonic games to computing a maximum $\{K_2, K_3\}$-packing. For a W-hedonic game (N, R), construct a graph $G = (N \cup (N \times \{0, 1\}), E)$ such that $\{(i, 0), (i, 1)\} \in E$ for all $i \in N$; $\{i, j\} \in E$ if and only if $i \in \max_{R_j}(N)$ and $j \in \max_{R_i}(N)$ for $i, j \in N$ such that $i \neq j$; and $\{i, (i, 0)\}, \{i, (i, 1)\} \in E$ if and only if $i \in \max_{R_i}(N)$ for all $i \in N$. Let H be a maximum $\{K_2, K_3\}$-packing of G.

It can then be proved that there exists a perfect partition of N according to R if and only if $|V(H)| = 3|N|$. We omit the technical details due to space restrictions.

Since PP for W-hedonic games reduces to checking whether graph G can be packed perfectly by elements in $\mathscr{F} = \{K_2, K_3\}$, we have a polynomial-time algorithm to solve

[3] In fact, Abraham and Manlove [2004] state that 'the case where preference lists [...] may include ties merits further investigation.'

PP for W-hedonic games. Denote by $CC(H)$ the set of connected components of graph H. If $|V(H)| = 3|N|$ and a perfect partition does exist, then $\{V(S) \cap N : S \in CC(H)\} \setminus \emptyset$ is a perfect partition. □

Similarly, the following is evident from the arguments in the proof of Theorem 7.

Theorem 8 *For W-hedonic games, it can be checked in polynomial time whether a given partition is Pareto optimal or weakly Pareto optimal.*

Our positive results for W-hedonic games also apply to hedonic games with $\mathscr{W}$-preferences.

5.4 B-hedonic games

We saw that for W-hedonic games, a Pareto optimal partition can be computed efficiently, even in the presence of unacceptable players. In the absence of unacceptable players, computing a Pareto optimal and individually rational partition is trivial in B-hedonic games, as the partition consisting of the grand coalition is a solution. Interestingly, if preferences do allow for unacceptable players, the same problem becomes NP-hard.

Theorem 9 *For B-hedonic games, computing a Pareto optimal partition is NP-hard.*

Proof sketch: It can be checked in polynomial time whether a partition is perfect in a B-hedonic game. Hence, by Lemma 1, it suffices to show that PP is NP-hard. We do so by a reduction from Sat. Let $\varphi = X_1 \wedge \cdots \wedge X_k$ a Boolean formula in conjunctive normal form in which the Boolean variables $p_1, \ldots, p_m$ occur. Now define the B-hedonic game (N, R), where $N = \{X_1, \ldots, X_k\} \cup \{p_1, \neg p_1, \ldots, p_m, \neg p_m\} \cup \{0, 1\}$ and the preferences for each literal p or $\neg p$, and each clause $X = (x_1 \vee \cdots \vee x_\ell)$ are denoted by lists of equivalence classes of equally preferred players in decreasing order of preference, as follows,

$$\begin{aligned} p: &\quad \{0,1\},\ N \setminus \{0,1,\neg p\},\ \{\neg p\} \\ \neg p: &\quad \{0,1\},\ N \setminus \{0,1,p\},\ \{p\} \\ X: &\quad \{x_1,\ldots,x_\ell\},\ N \setminus \{0,x_1,\ldots,x_\ell\},\ \{0\} \\ 0: &\quad N \setminus \{0,1\},\ \{0\},\ \{1\} \\ 1: &\quad N \setminus \{0,1\},\ \{1\},\ \{0\} \end{aligned}$$

We prove that φ is satisfiable if and only if a perfect (and individually rational) partition for (N, R) exists. The proof details are omitted due to space limitations. □

By using similar techniques, the following can be proved.

Theorem 10 *For B-hedonic games, verifying whether a partition is weakly Pareto optimal is coNP-complete.*

6 Conclusions

Pareto optimality and individual rationality are important requirements for desirable partitions in coalition formation. In this paper, we examined computational and structural issues related to Pareto optimality in various classes of hedonic games (see Table 1). We saw that unacceptability and ties are a major source of intractability when computing Pareto optimal outcomes. In some cases, checking whether a given partition is Pareto optimal can be significantly harder than finding one. We expect Theorem 10 to also hold for Pareto optimality instead of weak Pareto optimality.

Game	Verification	Computation
General	coNP-C (Th. 4)	NP-hard (Th. 3)
General (strict)	coNP-C (Th. 4)	in P (Prop. 1)
Roommate	in P (Th. 6)	in P (Th. 5)
B-hedonic	coNP-C (Th. 10, weak PO)	NP-hard (Th. 9)
W-hedonic	in P (Th. 8)	in P (Th. 7)

Table 1: Complexity of Pareto optimality in hedonic games: positive results hold for both Pareto optimality and individual rationality.

It should be noted that most of our insights gained into Pareto optimality and the resulting algorithmic techniques—especially those presented in Section 3 and Section 4—do not only apply to coalition formation but to any discrete allocation setting.

References

A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–702, 1998.

D. J. Abraham and D. F. Manlove. Pareto optimality in the roommates problem. Technical Report TR-2004-182, University of Glasgow, Department of Computing Science, 2004.

C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.

S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18:135–153, 2001.

A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

K. Cechlárová and J. Hajduková. Stability of partitions under $\mathscr{BW}$-preferences and $\mathscr{WB}$-preferences. *International Journal of Information Technology and Decision Making*, 3(4):605–618, 2004.

K. Cechlárová and J. Hajduková. Stable partitions with $\mathscr{W}$-preferences. *Discrete Applied Mathematics*, 138(3):333–347, 2004.

G. Cornuéjols, D. Hartvigsen, and W. Pulleyblank. Packing subgraphs in a graph. *Operations Research Letters*, 1(4):139–143, 1982.

J. H. Drèze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4):987–1003, 1980.

H. N. Gabow and R. Tarjan. Faster scaling algorithms for general graph matching problems. *Journal of the ACM*, 38(4):815–853, October 1991.

P. Hell and D. G. Kirkpatrick. Packings by cliques and by finite families of graphs. *Discrete Mathematics*, 49(1):45–59, 1984.

T. Morrill. The roommates problem revisited. *Journal of Economic Theory*, 145(5):1739–1756, 2010.

E. Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11:285–304, 1990.

Probabilistic and Utility-theoretic Models in Social Choice: Challenges for Learning, Elicitation, and Manipulation

Craig Boutilier
University of Toronto
Department of Computer Science
cebly@cs.toronto.edu

Tyler Lu
University of Toronto
Department of Computer Science
tl@cs.toronto.edu

1 Introduction

The abundance of inexpensive preference data facilitated by online commerce, search, recommender systems, and social networks has the potential to stretch the boundaries of social choice. Specifically, concepts and models usually applied to high stakes domains such as political elections, public or corporate policy decisions, and the like, will increasingly find themselves used in the lower stakes, high-frequency domains addressed by online systems.

Here are just two of many examples that are not typically interpreted as social choice problems, but which in fact, can profitably viewed as such. First, consider the (first page of) results returned by your favorite search engine to a specific query. While pure personalization, taking into account your specific preferences for results, would be ideal, this is generally not possible because of data scarcity. Hence the small amount of information known about your preferences is aggregated with the (equally scarce) data about users similar to you to determine the best results. This is a consensus decision making problem, since a single set of results is constructed for a collection of users, each of whom may have somewhat different preferences. Indeed, within the subfield of rank learning within machine learning, the *label ranking* paradigm [4] makes this assumption explicit. As a second example, consider the problem of an online retailer determining which *subset of size* k of potential products to offer to its target market. Ideally, the retailer would *segment* its audience into k groups such that a single product would be desirable to each member of the group [6]. Again, since a single choice is being proposed for all members of the group, this is a social choice problem [7].

Several factors make these and related problems both interesting and rather novel from a social choice perspective. First, the expression of complete preferences is wildly impractical: users will simply not tolerate much in the way of elicitation; and typically preferences will be *estimated* from choice behavior, partial ratings data, etc. Second, massive amounts of such data will in fact make it feasible to learn quite compelling probabilistic models of user preferences. Third, approximation will be an absolute necessity for several reasons: the need for "nearly instantaneous" recommendations will demand computational approximation; the incompleteness of preference data will demand informational approximation; and finally, very clear (usually economic) tradeoffs can be made that greatly facilitate the design of approximation methods (unlike, say, in political elections, where an "approximate winner" is unlikely to be viewed as satisfactory).

Issues of computational approximation have been studied extensively in social choice; informational approximation (dealing with incomplete preferences) has been too (though to a lesser extent); and probabilistic models have been used in analysis.[1] However, we feel the new demands of online systems call for a different style of analysis of social choice models and algorithms. Two key components lie at the heart of our proposal for such analyses: (a) utility-theoretic approximation, be it informational or computational; and (b) learning and exploiting probabilistic models of user preferences. We outline four broad categories of research challenges based on these components.

In what follows, we use A to denote a set of alternatives; U, a set of users or voters; v, a ranking, permutation, or *vote* over A; V, the set of permutations; $\mathbf{v}$, a *profile* with one (ranked) vote per voter; and r a voting rule, with $r(\mathbf{v})$ denoting the selected alternative given $\mathbf{v}$.

2 Learning Preferences

By a *probabilistic model*, we simply mean some distribution P over the set of rankings (or preferences) V. We'll discuss below various ways to exploit probabilistic models of user preferences when tackling various problems in social choice. However, one first needs *realistic* models of user preferences that support tractable inference and can be effectively learned from readily available data. Analysis of voting schemes in social choice tends to focus on models such as impartial culture which have little connection to reality in the settings mentioned above (or even in electoral data [11]).[2]

A number of models have been developed in econometrics, statistics and psychometrics that explicitly try to reflect the processes by which human comparison judgements are made, and are used to model population preferences. It is impossible to do justice to this literature here [10], but several of these models—especially the Mallows and Plackett-Luce models—have been appropriated by the machine learn-

[1] In this short position paper, we unfortunately must exclude references, even representative ones, on these topics.

[2] And even then, the questions addressed using such models tend to be very different than those we outline below.

ing community under the guise of "learning to rank" (LeToR). This has precipitated the development of many interesting methods for tractable learning and probabilistic inference with such models. This work is vitally important for the application of computational social choice, and we believe a rapprochement between the two disciplines is in order.

Of course, the flow works in both directions: the problems that arise in social choice must influence the development of new models and algorithms for learning and probabilistic inference. As one example, most work in LeToR assumes that observed rankings are noisy estimates of some underlying objective ranking (rather than representing genuinely distinct preferences). Because of the types of data sets considered, several important problems have gone unaddressed. For example, learning Mallows models is widely considered to be intractable with choice data consisting of pairwise comparisons of form $a_i \succ a_j$, obviously an important form of evidence in any social choice problem. We've developed a new model that allows Mallows models (and mixtures thereof) to be effectively learned from such data [8]. At its heart is the *generalized repeated insertion model (GRIM)*, that that allows approximate sampling of rankings conditioned on pairwise evidence.[3] With several real-world data sets, we've learned interesting population models with this technique.

Of course, this is just a start. More general models that support effective inference and tractable learning are needed, especially models that are tuned to the types of preference distributions we expect to find in consensus decision making domains. For example, realistic, tractable models for distributions over single-peaked preferences seem largely to have been unaddressed (and the "riffle independence" concept developed in ML may prove useful [3]).

3 Optimization

A second key issue is critical in the design of social choice methods for online settings, centered on the notion of utility-theoretic approximation of recommendations or "winners," especially when we have partial information about user preferences. While incomplete preferences are studied in a variety of guises, little attention is paid to the question of how to *select* a winner in such a situation.[4] In recent work, we've proposed using the notion of *minimax regret (MMR)* for just this purpose [9].

Most voting rules can be defined using a natural *scoring function* $s(a, \mathbf{v})$ that measures the quality or utility of alternative a given profile $\mathbf{v}$, i.e., $r(\mathbf{v}) \in \operatorname{argmax}_{a \in A} s(a, \mathbf{v})$. Now suppose we have access only to partial votes of some of the voters; i.e., replace each vote v with a (possibly empty) partial order p, or a collection of pairwise comparisons. Let $\mathbf{p}$ denote this *partial profile*. How should one select a winner? Intuitively, we measure the quality of a given $\mathbf{p}$ by considering how far from optimal a could be in the worst case (i.e., given any *completion* or extension $\mathbf{v} \in C(\mathbf{p})$ of $\mathbf{p}$). The minimax optimal solution is any alternative that is nearest to optimal in the worst case. More formally:

$$\begin{aligned} Regret(a, \mathbf{v}) &= max_{a' \in A} s(a', \mathbf{v}) - s(a, \mathbf{v}) \\ MR(a, \mathbf{p}) &= max_{\mathbf{v} \in C(\mathbf{p})} Regret(a, \mathbf{v}) \\ MMR(\mathbf{p}) &= min_{a \in A} MR(a, \mathbf{p}) \\ a^*_{\mathbf{p}} &\in \operatorname*{argmin}_{a \in A} MR(a, \mathbf{p}) \end{aligned}$$

This is a natural robustness criterion: the minimax winner $a^*_{\mathbf{p}}$ provides us with the tightest possible bound on loss of "societal utility." MMR can be computed in polytime for a variety of voting rules, and can offer quite distinct recommendations compared to selecting among possible winners [9].

One might consider minimax regret to be too pessimistic, though we argue below that it is, in fact, a very effective driver of vote elicitation/active learning. MMR also fails to exploit distributional information P about voter preferences. With such a probabilistic model, one can instead select a winner by maximizing expected utility (MEU): $a^*_{\mathbf{p}} = \operatorname{argmax} \sum_{\mathbf{v}} P(\mathbf{v}|\mathbf{p}) s(a, \mathbf{v})$. The investigation of algorithms for solving this computationally challenging problem for various combinations of voting rules and preference distributions is, in our opinion, a vital direction.

Notice that MEU ensures (Bayesian) optimality in the presence of a partial profile, but provides no guidance w.r.t. potential loss relative to choosing a winner with a complete profile $\mathbf{v}$. This stands in contrast to MMR, which tells us the potential value of adding new evidence to complete the vote profile. In the probabilistic case, *expected regret* is the most natural measure of loss regarding a proposed alternative a: $ER(a, \mathbf{p}) = \sum_{\mathbf{v}} P(\mathbf{v}|\mathbf{p}) Regret(a, \mathbf{v})$.[5] Notice, of course, that the same alternative $a^*_{\mathbf{p}}$ maximizes expected utility and minimizes expected regret; but ER is much more informative and useful for elicitation purposes.

4 Elicitation

Preference/vote elicitation is another critical process that has received insufficient attention in social choice. By explicitly articulating a notion of "societal" utility, and developing suitable probabilistic models, natural approaches to elicitation emerge that exploit the optimization criteria discussed above. Connections to active learning also become much clearer when adopting this perspective.

Without a probabilistic model P, MMR is probably the most natural criterion for robust selection of alternatives. But if MMR is too great, the potential error associated with *any* winner will be unacceptable. MMR can be reduced by asking some voter(s) some query(ies) about their preferences. In [9] we developed elicitation schemes that exploit the current solution to the minimax problem to determine appropriate voter-query pairs: on both synthetic and real-world voting and preference data, these methods performed extremely well, asking only a fraction of the queries that would be require to fully elicit voter rankings.[6] This is true despite the rather pessimistic worst-case results on the communication complexity of many voting rules. MMR also provides strong, distribution-free quality guarantees.

[3] This generalizes the *repeated insertion model* [2] for unconditional Mallows sampling.

[4] Necessary and possible winners don't actually prescribe general methods for selection.

[5] See Smith [12] who uses score-based regret.

[6] See Kalech et al. [5] for an alternative approach to elicitation.

In the probabilistic case, expected regret is the appropriate measure of loss, and optimal queries are those with maximum *expected value of information (EVOI)*. EVOI can be very difficult to compute in general, so again, as with MEU and ER computation, interesting challenges lay ahead in the effective (possibly approximate) computation of EVOI for various families of distributions and voting rules.

Interestingly, there are very useful ways of combining the probabilistic and regret-based perspectives. One difficulty with vote elicitation is that it is unrealistic to expect a fully interactive approach: no user u will want to answer a query, then wait for other users to answer their queries before the system returns with the next query for u. There is a fundamental tradeoff between amount of information elicited and the number of "query rounds" [5]. Probabilistic models can be used to help *batch* queries to assess this tradeoff. For instance, given a voting rule and a distribution, we may ask about the impact of asking m random users a small set of queries, e.g., "what are your top t alternatives?" For any t we can assess the posterior distribution over either MMR or ER to determine the depth t that makes the right tradeoff. That is, for given voting rules and families of distributions, we'd like effective techniques to compute, say, $E_P[MMR(\mathbf{p})|m,t]$, where expectation is taken over possible responses to the top-t queries from m users. Alternatively, one might favor a PAC-style analysis, deriving appropriate values for m and t such that $P(MMR(\mathbf{p}) < \varepsilon) \geq 1-\delta$: in other words, for the selected m and t, with high probability $1-\delta$, MMR will be less than some small value ε if we ask m voters for their top-t candidates. Analysis of this type (for various classes of queries) can be used to drastically limit the number of rounds while keeping the total amount of elicited information small.[7]

5 Manipulation

Finally, we close by suggesting that the utility-theoretic and probabilistic perspectives can provide a much more nuanced analysis of manipulation. Most manipulation analysis addresses the question of whether a small coalition of voters can change the outcome of an election by misreporting their preferences under *some distribution* of the preferences of the electorate. Typically, this distribution is a point distribution in which the coalition knows the exact preferences of other voters. Probabilistic information is sometimes used, but usually only to analyze the odds that a manipulation exists *assuming complete knowledge* on the part of the manipulators.

We suggest that two different styles of analysis would be much more useful when considering the application of social choice in the domains described above. First, assuming that manipulators know the full preference profile is unrealistic. Of course, it would be equally unrealistic to assume no knowledge: instead we suggest that analyses should *restrict* the manipulators' knowledge in reasonable ways. For example, we may insist that the distribution over preferences known to the manipulators has some minimum entropy; or we could restrict knowledge of preferences to that obtainable using a small number of samples from the underlying distribution. Such analysis of the potential for manipulation should also be undertaken using realistic distributions of preferences as opposed to impartial culture and related models.

The second change in analysis is suggested by the use of societal utility measures. Intuitively, if a *small* coalition can change the outcome from the *true* winner a to an alternative b, then it is highly likely b had a reasonably high societal utility to begin with. So rather than asking whether specific voting rules are manipulable, we can instead ask how much "damage" can a small coalition do: in other words, what is the maximum regret $MR(b,\mathbf{p})$ or expected regret $ER(b,\mathbf{p})$ given partial knowledge $\mathbf{p}$ obtained by the manipulators. The susceptibility of a voting rule to manipulation can then be characterized by placing limits on the form of $\mathbf{p}$, maximizing these damage metrics over possible manipulations b, and maximizing or taking expectation w.r.t. $\mathbf{p}$ of some limited form. Here is just one concrete question of this form: given distribution P, what is $E_{\mathbf{p}[m]\sim P} \max_b ER(b,\mathbf{p})$, where $\mathbf{p}[m] \sim P$ refers to random sample of m votes from P. This type of analysis may provide a very different view of the manipulability of various voting rules.

Acknowledgements: Thanks to Yann Chevaleyre, J erôme Lang, and Nicolas Maudet for very engaging discussions on several of these broad topics (and some of the specific problems mentioned here).

[7] Preliminary results suggest that reasonable bounds can be derived for Borda scoring with Mallows models. Some relevant results on sorting complexity for Mallows models are developed in [1].

References

[1] M. Braverman and E. Mossel. Sorting from noisy information. Manuscript, arXiv:0910.1191, 2009.

[2] J. P. Doignon, A. Pekec, and M. Regenwetter. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika*, 69(1):33–54, 2004.

[3] J. Huang and C. Guestrin. Riffled independence for ranked data. *NIPS 21*, pp.799–807, Vancouver, 2009.

[4] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.

[5] M. Kalech, S. Kraus, G. Kaminka and C. Goldman Practical voting rules with partial information. *Journal of Autonomous Agents and Multi-Agent Systems*, 22:151–182, 2011.

[6] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. *Journal of the ACM*, 51:263–280, 2004.

[7] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. *IJCAI-11*, Barcelona, 2011. To appear.

[8] T. Lu and C. Boutilier. Learning mallows models with pairwise preferences. *ICML-11*, Bellevue, WA, 2011. To appear.

[9] T. Lu and C. Boutilier. Robust approximation and incremental elicitation in voting protocols. *IJCAI-11*, Barcelona, 2011. To appear.

[10] J. Marden. *Analyzing and modeling rank data*. Chapman and Hall, 1995.

[11] M. Regenwetter, B. Grofman, A. A. J. Marley, and I. Tsetlin. *Behavioral Social Choice: Probabilistic Models, Statistical Inference, and Applications*. Cambridge University Press, 2006.

[12] W. Smith. Range voting. http://www.math.temple.edu/~wds/homepage/rangevote.pdf, 2000.

Necessary and Sufficient Conditions for the Strategyproofness of Irresolute Social Choice Functions

Felix Brandt and Markus Brill
Technische Universität München
85748 Garching bei München, Germany
{brandtf,brill}@in.tum.de

Abstract

While the Gibbard-Satterthwaite theorem states that every non-dictatorial and resolute, i.e., single-valued, social choice function is manipulable, it was recently shown that a number of appealing irresolute Condorcet extensions are strategyproof according to Kelly's preference extension. In this paper, we study whether these results carry over to stronger preference extensions due to Fishburn and Gärdenfors. For both preference extensions, we provide sufficient conditions for strategyproofness and identify social choice functions that satisfy these conditions, answering a question by Gärdenfors (1976) in the affirmative. We also show that some more discriminatory social choice functions fail to satisfy necessary conditions for strategyproofness.

1 Introduction

One of the central results in social choice theory states that every non-trivial social choice function (SCF)—a function mapping individual preferences to a collective choice—is susceptible to strategic manipulation (Gibbard, 1973; Satterthwaite, 1975). However, the classic result by Gibbard and Satterthwaite only applies to *resolute*, i.e., single-valued, SCFs. This assumption has been criticized for being unnatural and unreasonable (Gärdenfors, 1976; Kelly, 1977). As Taylor (2005) puts it, "If there is a weakness to the Gibbard-Satterthwaite theorem, it is the assumption that winners are unique." For example, consider a situation with two agents and two alternatives such that each agent prefers a different alternative. The problem is not that a resolute SCF has to select a single alternative (which is a well-motivated practical requirement), but that it has to select a single alternative *based on the individual preferences alone* (see, e.g., Kelly, 1977). As a consequence, the SCF has to be biased towards an alternative or a voter (or both). Resoluteness is therefore at variance with such elementary fairness notions as neutrality (symmetry among the alternatives) and anonymity (symmetry among the voters).

In order to remedy this shortcoming, Gibbard (1977) went on to characterize the class of strategyproof *decision schemes*, i.e., aggregation functions that yield probability distributions over the set of alternatives rather than single alternatives (see also Gibbard, 1978; Barberà, 1979). This class consists of rather degenerate decision schemes and Gibbard's characterization is therefore commonly interpreted as another impossibility result. However, Gibbard's theorem rests on unusually strong assumptions with respect to the voters' preferences. In contrast to the traditional setup in social choice theory, which typically only involves ordinal preferences, his result relies on the axioms of von Neumann and Morgenstern (1947) (or an equivalent set of axioms) in order to compare lotteries over alternatives. The gap between Gibbard and Satterthwaite's theorem for resolute SCFs and Gibbard's theorem for decision schemes has been filled by a number of impossibility results with varying underlying notions of how to compare sets of alternatives with each other (e.g., Gärdenfors, 1976; Barberà, 1977a,b; Kelly, 1977; Duggan and Schwartz, 2000; Barberà et al., 2001; Ching and Zhou, 2002; Sato, 2008; Umezawa, 2009), many of which are surveyed by Taylor (2005) and Barberà (2010).

How preferences over sets of alternatives relate to or depend on preferences over individual alternatives is a fundamental issue that goes back to at least de Finetti (1937) and Savage (1954). In the context of social choice the alternatives are usually interpreted as mutually exclusive candidates for a unique final choice. For instance, assume an agent prefers a to b, b to c, and—by transitivity—a to c. What can we reasonably deduce from this about his preferences over the subsets of $\{a, b, c\}$? It stands to reason to assume that he would strictly prefer $\{a\}$ to $\{b\}$, and $\{b\}$ to $\{c\}$. If a single alternative is eventually chosen using a procedure that is beyond the agent's control, it is safe to assume that he also prefers $\{a\}$ to $\{b, c\}$ (Kelly's extension), but whether he prefers $\{a, b\}$ to $\{a, b, c\}$ already depends on (his knowledge about) the final decision process. In the case of a lottery over all pre-selected alternatives according to a known *a priori* probability distribution with full support, he would prefer $\{a, b\}$ to $\{a, b, c\}$ (Fishburn's extension). This assumption is, however, not sufficient to separate $\{a, b\}$ and $\{a, c\}$. Based on a sure-thing principle which prescribes that alternatives present in both choice sets can be ignored, it would be natural to prefer the former to the latter (Gärdenfors' extension). Finally, whether the agent prefers $\{a, c\}$ to $\{b\}$ depends on his attitude towards risk: he might hope for his most-preferred alternative (leximax extension), fear that his worst alternative will be chosen

(leximin extension), or maximize his expected utility.

In general, there are at least three interdependent reasons why it is important to get a proper conceptual hold and a formal understanding of how preferences over sets relate to preferences over individual alternatives.

Rationality constraints. The examples above show that depending on the situation that is being modeled, preferences over sets are subject to certain rationality constraints, even if the preferences over individual alternatives are not. Not taking this into account would obviously be detrimental to a proper understanding of the situation at hand.

Epistemic and informational considerations. In many applications preferences over all subsets may be unavailable, unknown, or at least harder to obtain than preferences over the individual alternatives. With a proper grasp of how set preferences relate to preferences over alternatives, however, one may still be able to extract important structural information about the set preferences. In a similar vein, agents may not be fully informed about the situation they are in, e.g., they may not know the kind of lottery by means of which final choices are selected from sets. The less the agents know about the selection procedure, the less may be assumed about the structural properties of their preferences over sets.

Succinct representations. Clearly, as the set of subsets grows exponentially in the number of alternatives, preferences over subsets become prohibitively large. Hence, explicit representation and straightforward elicitation are not feasible and the succinct representation of set preferences becomes inevitable. Preferences over individual alternatives are of linear size and are the most natural basis for any succinct representation. Even when preferences over sets are succinctly represented by more elaborate structures than just preferences over individual alternatives, having a firm conceptual grasp on how set preferences relate to preferences over single alternatives is of crucial importance.

Any function that yields a preference relation over subsets of alternatives when given a preference relation over individual alternatives is called a *preference extension* or *set extension*. How to extend preferences to subsets is a fundamental issue that pervades the mathematical social sciences and has numerous applications in a variety of its disciplines. One example given by Gärdenfors (1979) is the following: "suppose one only has ordinal information about the welfare of the members of society. When is it possible to say that one *group* of people is better off than another group?"

In this paper, we will be concerned with three of the most well-known preference extensions due to Kelly (1977), Fishburn (1972), and Gärdenfors (1976). On the one hand, we provide sufficient conditions for strategyproofness and identify social choice functions that satisfy these conditions. For example, we show that the top cycle is strategyproof according to Gärdenfors' set extension, answering a question by Gärdenfors (1976) in the affirmative. On the other hand, we propose necessary conditions for strategyproofness and show that some more discriminatory social choice functions such as the minimal covering set and the bipartisan set, which have recently been shown to be strategyproof according to Kelly's extension, fail to satisfy strategyproofness according to Fishburn's and Gärdenfors' extension. By means of a counterexample, we also show that Gärdenfors (1976) incorrectly claimed that the SCF that returns the Condorcet winner when it exists and all Pareto-undominated alternatives otherwise is strategyproof according to Gärdenfors' extension.

2 Preliminaries

In this section, we provide the terminology and notation required for our results.

2.1 Social Choice Functions

Let $N = \{1, \dots, n\}$ be a set of voters with preferences over a finite set A of alternatives. The preferences of voter $i \in N$ are represented by a complete and *anti-symmetric* preference relation $R_i \subseteq A \times A$.[1] We have $a\ R_i\ b$ denote that voter i values alternative a at least as much as alternative b. In accordance with conventional notation, we write P_i for the strict part of R_i, i.e., $a\ P_i\ b$ if $a\ R_i\ b$ but not $b\ R_i\ a$. As R_i is anti-symmetric, $a\ P_i\ b$ if and only if $a\ R_i\ b$ and $a \neq b$. The set of all preference relations over A will be denoted by $\mathcal{R}(A)$. The set of *preference profiles*, i.e., finite vectors of preference relations, is then given by $\mathcal{R}^*(A)$. The typical element of $\mathcal{R}^*(A)$ will be $R = (R_1, \dots, R_n)$.

The following notational convention will turn out to be useful. For a given preference profile R with $b\ R_i\ a$, $R_{i:(a,b)}$ denotes the preference profile

$$(R_1, \dots, R_{i-1}, R_i \setminus \{(b,a)\} \cup \{(a,b)\}, R_{i+1}, \dots, R_n).$$

That is, $R_{i:(a,b)}$ is identical to R except that alternative a is strengthened with respect to b within voter i's preference relation.

Our central object of study are *social choice functions*, i.e., functions that map the individual preferences of the voters to a non-empty set of socially preferred alternatives.

Definition 1. A *social choice function (SCF)* is a function $f : \mathcal{R}^*(A) \to 2^A \setminus \emptyset$.

An SCF f is said to be based on pairwise comparisons (or simply *pairwise*) if, for all preference profiles R and R', $f(R) = f(R')$ whenever for all alternatives a, b,

$$\begin{aligned} &|\{i \in N \mid a\ R_i\ b\}| - |\{i \in N \mid b\ R_i\ a\}| \\ &= |\{i \in N \mid a\ R'_i\ b\}| - |\{i \in N \mid b\ R'_i\ a\}|. \end{aligned}$$

In other words, the outcome of a pairwise SCF only depends on the comparisons between pairs of alternatives (see, e.g., Young, 1974; Zwicker, 1991).

[1] For most of our results, we do not assume transitivity of preferences. In fact, Theorems 3 and 5 become stronger but are easier to prove for general—possibly intransitive—preferences. Theorems 4 and 6, on the other hand, become slightly weaker because there exist SCFs that are only manipulable if intransitive preferences are allowed. For all the manipulable SCFs in this paper, however, we show that they are manipulable even if transitive preferences are required.

For a given preference profile $R = (R_1, \ldots, R_n)$, the *majority relation* $R_M \subseteq A \times A$ is defined by $a\ R_M\ b$ if and only if $|\{i \in N \mid a\ R_i\ b\}| \geq |\{i \in N \mid b\ R_i\ a\}|$. Let P_M denote the strict part of R_M. A *Condorcet winner* is an alternative a that is preferred to any other alternative by a strict majority of voters, i.e., $a\ P_M\ b$ for all alternatives $b \neq a$. An SCF is called a *Condorcet extension* if it uniquely selects the Condorcet winner whenever one exists.

We will now introduce the SCFs considered in this paper. With the exception of the Pareto rule and the omninomination rule, all of these SCFs are pairwise Condorcet extensions.

Pareto rule An alternative a is *Pareto-dominated* if there exists an alternative b such that $b\ P_i\ a$ for all voters $i \in N$. The Pareto rule PAR returns all alternatives that are *not* Pareto-dominated.

Omninomination rule The omninomination rule $OMNI$ returns all alternatives that are ranked first by at least one voter.

Condorcet rule The Condorcet rule $COND$ returns the Condorcet winner if it exists, and all alternatives otherwise.

Top Cycle Let R_M^* denote the transitive closure of the majority relation, i.e., $a\ R_M^*\ b$ if and only if there exists $k \in \mathbb{N}$ and $a_1, \ldots, a_k \in A$ with $a_1 = a$ and $a_k = b$ such that $a_i\ R_M\ a_{i+1}$ for all $i < k$. The top cycle rule TC (also known as *weak closure maximality*, *GETCHA*, or the *Smith set*) returns the maximal elements of R_M^*, i.e., $TC(R) = \{a \in A \mid a\ R_M^*\ b \text{ for all } b \in A\}$ (Good, 1971; Smith, 1973; Schwartz, 1986).

Minimal Covering Set A subset $C \subseteq A$ is called a *covering set* if for all alternatives $b \in A \setminus C$, there exists $a \in C$ such that $a\ P_M\ b$ and for all $c \in C \setminus \{a\}$, $b\ P_M\ c$ implies $a\ P_M\ c$ and $c\ P_M\ a$ implies $c\ P_M\ b$. Dutta (1988) and Dutta and Laslier (1999) have shown that there always exists a unique *minimal* covering set. The SCF MC returns exactly this set.

Bipartisan Set Consider the two-player zero-sum game in which the set of actions for both players is given by A and payoffs are defined as follows. If the first player chooses a and the second player chooses b, the payoff for the first player is 1 if $a\ P_M\ b$, -1 if $b\ P_M\ a$, and 0 otherwise. The bipartisan set BP contains all alternatives that are played with positive probability in some Nash equilibrium of this game (Laffond et al., 1993; Dutta and Laslier, 1999).

Observe that PAR and $OMNI$ are only well-defined for transitive individual preferences. It is well-known that $BP(R) \subseteq MC(R) \subseteq TC(R) \subseteq COND(R)$ for all preference profiles R. Furthermore, $MC(R) \subseteq PAR(R)$ and $OMNI(R) \subseteq PAR(R)$ for all R, but the choice sets of $OMNI$ and $COND$ may be disjoint.

2.2 Strategyproofness

An SCF is manipulable if one or more voters can misrepresent their preferences in order to obtain a more preferred choice set. While comparing choice set is trivial for resolute SCFs, this is not the case for irresolute ones. Whether one choice set is preferred to another depends on how the preferences over individual alternatives are to be extended to sets of alternatives.

In our investigation of strategyproof SCFs, we will consider the following three well-known set extensions due to Kelly (1977), Fishburn (1972),[2] and Gärdenfors (1976). Let R_i be a preference relation over A and $X, Y \subseteq A$.

- $X\ R_i^K\ Y$ if and only if $x\ R_i\ y$ for all $x \in X$ and all $y \in Y$ (Kelly, 1977)
  One interpretation of this extension is that voters are unaware of the lottery that will be used to pick the winning alternative (Gärdenfors, 1979).
- $X\ R_i^F\ Y$ if and only if $x\ R_i\ y$, $x\ R_i\ z$, and $y\ R_i\ z$ for all $x \in X \setminus Y$, $y \in X \cap Y$, and $z \in Y \setminus X$ (Fishburn, 1972)
  One interpretation of this extension is that voters are unaware of the *a priori* distribution underlying the lottery that picks the winning alternative (Ching and Zhou, 2002). Alternatively, one may assume the existence of a tie-breaker with linear, but unknown, preferences.
- $X\ R_i^G\ Y$ if and only if one of the following conditions is satisfied (Gärdenfors, 1976):
  (*i*) $X \subset Y$ and $x\ R_i\ y$ for all $x \in X$ and $y \in Y \setminus X$
  (*ii*) $Y \subset X$ and $x\ R_i\ y$ for all $x \in X \setminus Y$ and $y \in Y$
  (*iii*) neither $X \subset Y$ nor $Y \subset X$ and $x\ R_i\ y$ for all $x \in X \setminus Y$ and $y \in Y \setminus X$

  No interpretation in terms of lotteries is known for this set extension. Gärdenfors (1976) motivates it by alluding to Savage's sure-thing principle (when comparing two options, identical parts may be ignored). Unfortunately, the definition of this extension is somewhat "discontinuous," which is also reflected in the hardly elegant characterization given in Theorem 5.

It is easy to see that these extensions form an inclusion hierarchy.

Fact 1. *For all preference relations R_i and subsets $X, Y \subseteq A$,*

$$X\ R_i^K\ Y \text{ implies } X\ R_i^F\ Y \text{ implies } X\ R_i^G\ Y.$$

For $\mathcal{E} \in \{K, F, G\}$, let $P_i^{\mathcal{E}}$ denote the strict part of $R_i^{\mathcal{E}}$. As R_i is anti-symmetric, so is $R_i^{\mathcal{E}}$. Therefore, we have $X\ P_i^{\mathcal{E}}\ Y$ if and only if $X\ R_i^{\mathcal{E}}\ Y$ and $X \neq Y$.

Definition 2. Let $\mathcal{E} \in \{K, F, G\}$. An SCF f is *$P^{\mathcal{E}}$-manipulable by a group of voters $C \subseteq N$* if there exist preference profiles R and R' with $R_j = R'_j$ for all $j \notin C$ such that

$$f(R')\ P_i^{\mathcal{E}}\ f(R) \text{ for all } i \in C.$$

An SCF is *$P^{\mathcal{E}}$-strategyproof* if it is not $P^{\mathcal{E}}$-manipulable by single voters. An SCF is *$P^{\mathcal{E}}$-group-strategyproof* if it is not $P^{\mathcal{E}}$-manipulable by any group of voters.

[2] Gärdenfors (1979) attributed this extension to Fishburn because it is the weakest extension that satisfies a certain set of axioms proposed by Fishburn (1972).

Fact 1 implies that P^G-group-strategyproofness is stronger than P^F-group-strategyproofness, which in turn is stronger than P^K-group-strategyproofness. Note that, in contrast to some related papers, we interprete preference extensions as fully specified (incomplete) preference relations rather than minimal conditions on set preferences.

3 Related Work

Barberà (1977a) and Kelly (1977) have shown independently that all non-trivial SCFs that are rationalizable via a quasi-transitive preference relation are P^K-manipulable. However, as witnessed by various other (non-strategic) impossibility results that involve quasi-transitive rationalizability (e.g., Mas-Colell and Sonnenschein, 1972), it appears as if this property itself is unduly restrictive. As a consequence, Kelly (1977) concludes his paper by contemplating that "one plausible interpretation of such a theorem is that, rather than demonstrating the impossibility of reasonable strategy-proof social choice functions, it is part of a critique of the regularity [rationalizability] conditions."

Strengthening earlier results by Gärdenfors (1976) and Taylor (2005), Brandt (2011a) showed that no Condorcet extension is P^K-strategyproof. The proof, however, crucially depends on strategic tie-breaking and hence does not work for strict preferences. For this reason, only preference profiles with strict, i.e., anti-symmetric, preferences are considered in the present paper.

Brandt (2011a) also provided a sufficient condition for P^K-group-strategyproofness. *Set-monotonicity* can be seen as an irresolute variant of Maskin-monotonicity (Maskin, 1999) and prescribes that the choice set is invariant under the weakening of unchosen alternatives.

Definition 3. An SCF f satisfies *set-monotonicity (SET-MON)* if $f(R_{i:(a,b)}) = f(R)$ for all preference profiles R, voters i, and alternatives a, b with $b \notin f(R)$.

Theorem 1 (Brandt, 2011a)**.** *Every SCF that satisfies SET-MON is P^K-group-strategyproof.*

Set-monotonicity is a demanding condition, but a handful of SCFs such as TC, MC, and BP are known to be set-monotonic. For the class of *pairwise* SCFs, this condition is also necessary, which shows that many well-known SCFs such as Borda's rule, Copeland's rule, Kemeny's rule, the uncovered set, and the Banks set are not P^K-group-strategyproof.

Theorem 2 (Brandt, 2011a)**.** *Every pairwise SCF that is P^K-group-strategyproof satisfies SET-MON.*

Strategyproofness according to Kelly's extension thus draws a sharp line within the space of SCFs as almost all established non-pairwise SCFs (such as plurality and all weak Condorcet extensions like Young's rule) are also known to be P^K-manipulable (see, e.g., Taylor, 2005).

The state of affairs for Gärdenfors' and Fishburn's extensions is less clear. Gärdenfors (1976) has shown that $COND$ and $OMNI$ are P^G-group-strategyproof. In an attempt to extend this result to more discriminatory SCFs, he also claimed that $COND \cap PAR$, which returns the Condorcet winner if it exists and all Pareto-undominated alternatives otherwise, is P^G-strategyproof. However, we show that this is not the case (Proposition 2). Gärdenfors concludes that "we have not been able to find any more decisive function which is stable [strategyproof] and satisfies minimal requirements on democratic decision functions." We show that TC is such a function (Corollary 1).

Apart from a theorem by Ching and Zhou (2002), which uses an unusually strong definition of strategyproofness, we are not aware of any characterization result using Fishburn's extension. Feldman (1979) has shown that the Pareto rule is P^F-strategyproof and Sanver and Zwicker (2010) have shown that the same is true for TC.

4 Results

This section contains our results. Most proofs are omitted due to the space constraint.

4.1 Necessary and Sufficient Conditions for Group-Strategyproofness

We first introduce a new property that requires that modifying preferences between chosen alternatives may only result in smaller choice sets. Set-monotonicity entails a condition called *independence of unchosen alternatives*, which states that the choice set is invariant under modifications of the preferences between unchosen alternatives. Accordingly, the new property will be called *exclusive independence of chosen alternatives*, where "exclusive" refers to the requirement that unchosen alternatives remain unchosen.

Definition 4. An SCF f satisfies *exclusive independence of chosen alternatives (EICA)* if $f(R') \subseteq f(R)$ for all pairs of preference profiles R and R' that differ only on alternatives in $f(R)$, i.e., $R_i|_{\{a,b\}} = R'_i|_{\{a,b\}}$ for all $i \in N$ and all alternatives a, b with $b \notin f(R)$.

It turns out that, together with SET-MON, this new property is sufficient for an SCF to be group-strategyproof according to Fishburn's preference extension.

Theorem 3. *Every SCF that satisfies SET-MON and EICA is P^F-group-strategyproof.*

For *pairwise* SCFs, the following weakening of EICA can be shown to be necessary for group-strategyproofness according to Fishburn's extension. It prescribes that modifying preferences among chosen alternatives does not result in a choice set that is a strict superset of the original choice set.

Definition 5. An SCF f satisfies *weak EICA* if $f(R) \not\subset f(R')$ for all pairs of preference profiles R and R' that differ only on alternatives in $f(R)$.

Theorem 4. *Every pairwise SCF that is P^F-group-strategyproof satisfies SET-MON and weak EICA.*

We now turn to P^G-group-strategyproofness. When comparing two sets, P^G differs from P^F only in the case when neither set is contained in the other. The following definition captures exactly this case.

Definition 6. An SCF f satisfies the *symmetric difference property (SDP)* if either $f(R) \subseteq f(R')$ or $f(R') \subseteq f(R)$ for all pairs of preference profiles R and R' such that $R_i|_{\{a,b\}} =$

$R'_i|_{\{a,b\}}$ for all $i \in N$ and all alternatives a, b with $a \in f(R) \setminus f(R')$ and $b \in f(R') \setminus f(R)$.

Theorem 5. *Every SCF that satisfies SET-MON, EICA, and SDP is P^G-group-strategyproof.*

As was the case for Fishburn's extension, a set of necessary conditions for pairwise SCFs can be obtained by replacing EICA with weak EICA.

Theorem 6. *Every pairwise SCF that is P^G-group-strategyproof satisfies SET-MON, weak EICA, and SDP.*

4.2 Consequences

We are now ready to study the strategyproofness of the SCFs defined in Section 2. It can be checked that $COND$ and TC satisfy SET-MON, EICA, and SDP and thus, by Theorem 5, are P^G-group-strategyproof.

Corollary 1. *$COND$ and TC are P^G-group-strategyproof.*

$OMNI$, PAR, and $COND \cap PAR$ satisfy SET-MON and EICA, but not SDP.

Corollary 2. *$OMNI$, PAR, and $COND \cap PAR$ are P^F-group-strategyproof.*

As $OMNI$, PAR, and $COND \cap PAR$ are not pairwise, the fact that they violate SDP does *not* imply that they are P^G-manipulable. In fact, it turns out that $OMNI$ is strategyproof according to Gärdenfors' extension, while PAR and $COND \cap PAR$ are not.

Proposition 1. *$OMNI$ is P^G-group-strategyproof.*

Proposition 2. *PAR and $COND \cap PAR$ are P^G-manipulable.*

Proof. Consider the following profile $R = (R_1, R_2, R_3, R_4)$.

R_1	R_2	R_3	R_4
c	c	a	a
d	d	b	b
b	a	c	c
a	b	d	d

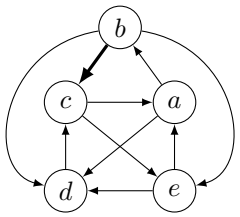
It is easily verified that $PAR(R) = \{a, b, c\}$. Now let $R' = (R'_1, R_2, R_3, R_4)$ where $R'_1 : d \succ c \succ a \succ b$. Obviously, $PAR(R') = \{a, c, d\}$ and $\{a, c, d\}\ P_1^G\ \{a, b, c\}$ because $d\ R_1\ b$. I.e., the first voter can obtain a preferable choice set by misrepresenting his preferences. As neither R nor R' has a Condorcet winner, the same holds for $COND \cap PAR$. □

Finally, we show that MC and BP violate weak EICA, which implies that both rules are manipulable according to Fishburn's extension.

Corollary 3. *MC and BP are P^F-manipulable.*

Proof. By Theorem 4 and the fact that both MC and BP are pairwise, it suffices to show that MC and BP violate weak EICA. To this end, consider the following profile $R = (R_1, R_2, R_3, R_4, R_5)$ and the corresponding majority graph representing P_M.

R_1	R_2	R_3	R_4	R_5
d	c	b	e	d
e	b	c	a	c
a	a	e	b	a
b	e	a	d	b
c	d	d	c	e



It can be checked that $MC(R) = BP(R) = \{a, b, c\}$. Define $R' = R_{1:(c,b)}$, i.e., the first voter strengthens c with respect to b. Observe that P_M and P'_M disagree on the pair $\{b, c\}$, and that $MC(R') = BP(R') = \{a, b, c, d, e\}$. Thus, both MC and BP violate weak EICA and the first voter can manipulate because $\{a, b, c, d, e\}\ P_1^F\ \{a, b, c\}$. □

The same example shows that the tournament equilibrium set (Schwartz, 1990) and the minimal extending set (Brandt, 2011b), both of which are only defined for an odd number of voters and conjectured to be P^K-group-strategyproof, are P^F-manipulable.

	P^K-str.pr.	P^F-str.pr.	P^G-str.pr.
$OMNI$	✓	✓	✓[a]
$COND$	✓	✓	✓[a]
TC	✓	✓[b]	✓
PAR	✓	✓[c]	–
$COND \cap PAR$	✓	✓	–
MC	✓	–	–
BP	✓	–	–

[a] Gärdenfors (1976)
[b] Sanver and Zwicker (2010)
[c] Feldman (1979)

Table 1: Summary of results.

5 Conclusion

In this paper, we investigated the effect of various preference extensions on the manipulability of irresolute SCFs. We proposed necessary and sufficient conditions for strategyproofness according to Fishburn's and Gärdenfors' set extensions and used these conditions to illuminate the strategyproofness of a number of well-known SCFs. Our results are summarized in Table 1. As mentioned in Section 3, some of these results were already known or—in the case of P^F-strategyproofness of the top cycle—have been discovered independently by other authors. In contrast to the papers by Gärdenfors (1976), Feldman (1979), and Sanver and Zwicker (2010), which more or less focus on particular SCFs, our axiomatic approach yields unified proofs of most of the statements in the table.[3]

Many interesting open problems remain. For example, it is not known whether there exists a Pareto-optimal pairwise

[3] The results in the leftmost column of Table 1 are due to Brandt (2011a) and are included for the sake of completeness.

SCF that is strategyproof according to Gärdenfors' extension. Recently, the study of the manipulation of irresolute SCFs by other means than untruthfully representing one's preferences—e.g., by abstaining the election (Pérez, 2001; Jimeno et al., 2009)—has been initiated. For the set extensions considered in this paper it is unknown which SCFs can be manipulated by abstention. It would be desirable to also obtain characterizations of these classes of SCFs and, more generally, to improve our understanding of the interplay between both types of manipulation. For instance, it is not difficult to show that the negative results in Corollary 3 also extend to manipulation by abstention.

Another interesting related question concerns the epistemic foundations of the above extensions. Most of the literature in social choice theory focusses on well-studied economic models where agents have full knowledge of a random selection process, which is often assumed to be a lottery with uniform probabilities. The study of more intricate distributed protocols or computational selection devices that justify certain set extensions appears to be very promising. For instance, Kelly's set extension could be justified by a distributed protocol for "unpredictable" random selections that do not permit a meaningful prior distribution.

References

S. Barberà. Manipulation of social decision functions. *Journal of Economic Theory*, 15(2):266–278, 1977a.

S. Barberà. The manipulation of social choice mechanisms that do not leave "too much" to chance. *Econometrica*, 45(7):1573–1588, 1977b.

S. Barberà. A note on group strategy-proof decision schemes. *Econometrica*, 47(3):637–640, 1979.

S. Barberà. Strategy-proof social choice. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 2, chapter 25, pages 731–832. Elsevier, 2010.

S. Barberà, B. Dutta, and A. Sen. Strategy-proof social choice correspondences. *Journal of Economic Theory*, 101(2):374–394, 2001.

F. Brandt. Group-strategyproof irresolute social choice functions. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011a. Forthcoming.

F. Brandt. Minimal stable sets in tournaments. *Journal of Economic Theory*, 2011b. Forthcoming.

S. Ching and L. Zhou. Multi-valued strategy-proof social choice rules. *Social Choice and Welfare*, 19:569–580, 2002.

B. de Finetti. La prévison: ses lois logiques, ses sources subjectives. *Annales de l'institut Henri Poincaré*, 7(1):1–68, 1937.

J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard-Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

B. Dutta. Covering sets and a new Condorcet choice correspondence. *Journal of Economic Theory*, 44:63–80, 1988.

B. Dutta and J.-F. Laslier. Comparison functions and choice correspondences. *Social Choice and Welfare*, 16(4):513–532, 1999.

A. Feldman. Manipulation and the Pareto rule. *Journal of Economic Theory*, 21:473–482, 1979.

P. C. Fishburn. Even-chance lotteries in social choice theory. *Theory and Decision*, 3:18–40, 1972.

P. Gärdenfors. Manipulation of social choice functions. *Journal of Economic Theory*, 13(2):217–228, 1976.

P. Gärdenfors. On definitions of manipulation of social choice functions. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.

A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41: 587–602, 1973.

A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–681, 1977.

A. Gibbard. Straightforwardness of game forms with lotteries as outcomes. *Econometrica*, 46(3):595–614, 1978.

I. J. Good. A note on Condorcet sets. *Public Choice*, 10:97–101, 1971.

J. L. Jimeno, J. Pérez, and E. García. An extension of the Moulin No Show Paradox for voting correspondences. *Social Choice and Welfare*, 33(3):343–459, 2009.

J. Kelly. Strategy-proofness and social choice functions without single-valuedness. *Econometrica*, 45(2):439–446, 1977.

G. Laffond, J.-F. Laslier, and M. Le Breton. The bipartisan set of a tournament game. *Games and Economic Behavior*, 5:182–201, 1993.

A. Mas-Colell and H. Sonnenschein. General possibility theorems for group decisions. *Review of Economic Studies*, 39(2):185–192, 1972.

E. Maskin. Nash equilibrium and welfare optimality. *Review of Economic Studies*, 66(26):23–38, 1999.

J. Pérez. The strong no show paradoxes are a common flaw in Condorcet voting correspondences. *Social Choice and Welfare*, 18 (3):601–616, 2001.

M. R. Sanver and W. S. Zwicker. Monotonicity properties and their adaption to irresolute social choice rules. Unpublished Manuscript, 2010.

S. Sato. On strategy-proof social choice correspondences. *Social Choice and Welfare*, 31:331–343, 2008.

M. A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10: 187–217, 1975.

L. J. Savage. *The Foundations of Statistics*. Wiley and Sons, 1954.

T. Schwartz. *The Logic of Collective Choice*. Columbia University Press, 1986.

T. Schwartz. Cyclic tournaments and cooperative majority voting: A solution. *Social Choice and Welfare*, 7:19–29, 1990.

J. H. Smith. Aggregation of preferences with variable electorate. *Econometrica*, 41(6):1027–1041, 1973.

A. D. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.

M. Umezawa. Coalitionally strategy-proof social choice correspondences and the Pareto rule. *Social Choice and Welfare*, 33:151–158, 2009.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 2nd edition, 1947.

H. P. Young. An axiomatization of Borda's rule. *Journal of Economic Theory*, 9:43–52, 1974.

W. S. Zwicker. The voter's paradox, spin, and the Borda count. *Mathematical Social Sciences*, 22:187–227, 1991.

Fairness and Welfare in Division of Goods When Utility is Transferable

Ruggiero Cavallo
Yahoo! Research
111 West 40th Street
New York, NY 10018
cavallo@yahoo-inc.com

Abstract

We join the goals of two giant and related fields of research in group decision-making whose connection has historically been underdeveloped: fair division, and efficient mechanism design with monetary payments. To do this we assume a context where utility is quasilinear and thus transferable across agents. We generalize the traditional binary criteria of envy-freeness, proportionality, and efficiency to measures of degree that range between 0 and 1. We observe the impossibility of achieving optimal social welfare with strategic agents in allocation of divisible or indivisible goods. We then set as the goal a strategyproof mechanism that achieves *high* welfare, *low* envy, and *low* disproportionality. We demonstrate that for the canonical fair division settings the VCG mechanism is typically not a satisfactory candidate, but the redistribution mechanism of [Bailey, 1997; Cavallo, 2006] is.

1 Introduction

The starting point in designing or evaluating any prospective group decision-making procedure is the question: what goals do we want to achieve? The answer of course will depend on the setting and who you ask. If individuals are selfish, then each will answer "maximize the value I get from the procedure". But this is usually a non-starter because, very often, what is optimal for one individual will be suboptimal for another. A goal that has a much more plausible chance of being endorsed by individuals in a group, selfish though they may be, is to achieve some notion of *fairness*. In settings that have a certain symmetric separability in the description of each outcome, we can consider notions such as *envy* and *proportionality*. Would any agent prefer the outcome obtained by another agent? Does each agent get at least a certain proportion of the value they would obtain if they could make the decision themselves, as a dictator?

These are exactly the fairness goals that have been taken up and formally studied by researchers in mathematics, economics, political science, and, most recently, computer science. The prototypical decision setting addressed in such work is that of *fair division*, where either a divisible good must be split up—typically analogized as a cake to be cut—or a set of indivisible goods is to be allocated amongst a set of stakeholders.

Perhaps the most basic and well-known example of a fair division procedure is the "you cut I choose" method for two agents: one agent determines a bisection (cuts the cake), and the other decides who gets which piece. This simple approach achieves the desirable properties of envy-freeness and proportionality: neither agent would prefer to swap pieces with the other, and both agents—in their own estimation—obtain at least half of the cake. Indeed, if we make no further assumptions about the agents it is difficult to see any way of improving on this approach. Yet, from a broader perspective we can see that a crucial aspect of the problem has been ignored: how *much* does each agent like cake? What if one of the agents' enjoyment (call her Alice) is only marginally improved from obtaining anything more than a small sliver, while the other (call him Bob) obtains only marginally increasing enjoyment until he obtains a very large portion? In such a situation, intuitively we feel it would be more just to "tip the scale" in favor of Bob, since his gain could be enormous while Alice's loss would be negligible for a skewed division.

We can formalize this intuition as a concern for *social welfare*. However, as intuitively basic as the concept is, the way we've described the setting so far does not allow us to consider it—there is a problem of comparing one agent's welfare to another's.[1] When Bob claims to have lower value for the same size piece of cake as Alice, how do we interpret that? The comparison becomes possible if we assume a quasilinear structure to agent utilities, as an agent's value for an allocation can then be interpreted as their "willingness to pay" for it. We can then also bring to bear the powerful tool of monetary payments: besides receiving a piece of cake, each agent can either be given money or have money taken away. The social welfare can then neatly and legitimately be defined as the sum of the agent utilities.

As we will see, even granting this quasilinear context, in general there will exist no mechanism that perfectly satisfies all three of our criteria: efficiency (i.e., full social welfare, defined as the social utility of the allocation that maximizes the

[1] And so the best we could aim for is a Pareto optimal allocation where no agent could benefit from a change that doesn't cause another to lose.

sum of agent values), envy-freeness, and proportionality. In fact, there can be no mechanism that yields full social welfare alone, because any unsubsidized efficient allocation requires the agents to make payments outside of the group. At the same time, although previous work in cake-cutting demonstrates the existence of perfectly envy-free and proportional allocations for arbitrary size groups [Neyman, 1946], feasible methods for determining such allocations are currently known only for groups of size less than 5. But the fact that procedures that perfectly satisfy our criteria don't exist is little reason to abandon hope. Instead, in this paper we pursue methods that, in expectation, obtain "good" performance along each metric—*high* social welfare, *low* envy, and *low* disproportionality for each agent.

1.1 Related work

We build on two significant bodies of literature: the fair division literature which typically assumes little to nothing about the nature of agent utility functions, and the mechanism design literature which, with few exceptions, has at its foundation the assumption of transferable, quasilinear utility.

Work in fair division, at least in a modern research context, seems to have been initiated by Steinhaus [1948] and Banach & Knaster (whom Steinhaus credits as discovering one of the foundational constructive approaches), who addressed the question of proportionality for groups of size greater than two. More recently Brams has been a key figure, providing, with coauthors, a series of procedures for obtaining envy-free allocations for 3 or 4 players that involve a limited number of "cuts" to the cake (see the text [Brams and Taylor, 1996]).

Also recently, the question of *truthfulness* has been introduced in this context—can an agent gain from misrepresenting his preferences about pieces of cake? Brams et al. [2006] consider a very limited kind of truthfulness, requiring for each agent only that there exist a case (i.e., preferences of other agents) where lying would not be beneficial. Chen et al. [2010] consider a much stronger and more compelling notion, the standard concept of *strategyproofness* from the social choice literature, wherein lying can never be beneficial regardless of the behavior of others; they propose a procedure that is strategyproof and proportional for restricted classes of value functions; Mossel and Tamuz [2010] address essentially the same problem.

Fairness has also been studied in a context of allocating *indivisible* goods (the "assignment problem"); the canonical example is "room assignment, rent division", where a group of housemates must divvy up the rooms in the house and decide what share of the rent is paid by whom. Brams and Kilgour [2001], Haake et al. [2002], and Abdulkadiroglu et al. [2004] all introduce efficient procedures that (in some cases) also achieve envy-freeness; however all simply assume truthful participation and break down in a context of strategic agents. This is perhaps unsurprising, as Alkan et al. [1991] earlier showed that there exists no envy-free and strategyproof mechanism (that is, without allowing for "extra" payments that diminish social welfare). In a similar spirit to the evaluation methodology we propose in the current paper, Lipton et al. [2004] consider measures of envy, and seek allocation procedures that are approximately envy-free.

Mechanism design (initiated Hurwicz [1960]) introduces payments as a way to obtain good outcomes in equilibrium when agents are self-interested and strategic. The hallmark positive result is the class of Groves mechanisms, wherein each agent reports a value function over outcomes, the socially optimal one is chosen, and each agent is paid the reported value of the others minus a constant. Green and Laffont [1977] and Holmstrom [1979] showed that this class exactly characterizes the efficient and strategyproof mechanisms for most practical problem domains. In settings where no outcome yields anyone negative value, the Vickrey–Clarke–Groves (VCG) mechanism [Vickrey, 1961; Clarke, 1971; Groves, 1973]—an instance of the Groves class where agents make payments commensurate with the negative externality they impose on others—additionally has the properties of *ex post individual rationality* and *no-deficit*: no agent is ever worse off from participating and aggregate payment to the agents is never positive.

Despite these attributes, in a group decision-making problem where the goal is welfare of the group, the VCG mechanism is unsatisfactory because it generates high *revenue*, payments that must be transferred outside the group and thus detract from social welfare. *Redistribution mechanisms*, introduced by Bailey [1997] and Cavallo [2006],[2] address this issue by returning large portions of VCG revenue back to the agents in a way that does not violate strategyproofness. Subsequently Guo and Conitzer [2007] and Moulin [2009] provided a mechanism for the special case of multi-unit auctions that maximizes the *worst-case* social welfare in that context.

Studies of the fairness properties of strategyproof mechanisms has mainly been confined to VCG. Exceptions are [Papai, 2003], which characterizes the set of all envy-free Groves mechanisms (i.e., all strategyproof, efficient, and envy-free mechanisms); and [Moulin, 2010], which examines an efficiency/fairness tradeoff in single-item allocation. In the assignment problem setting, Leonard [1983] showed that VCG is envy-free; Cohen et al. [2010] recently extended this result to a generalization of the assignment problem where individuals have additive value for obtaining more than one good.

Finally, like the current paper, [Porter *et al.*, 2004] also straddles the fair division and mechanism design literatures, there seeking to equitably allocate costly tasks throughout a population (see also [Moulin, 2010]). Interestingly, for the case of single-item allocation the mechanism earlier introduced in [Bailey, 1997] and later generalized in [Cavallo, 2006] is proposed.

1.2 Summary of contributions

Our first step in this paper will be to generalize the notions of efficiency (welfare), envy-freeness, and proportionality from the strict "yes or no" conception to *degrees*. So, for instance, given a probability distribution over types a mechanism may yield social welfare that is close to opti-

[2] Bailey was the first, to my knowledge, to derive a redistribution mechanism; his approach applies to single-item auctions as well as some other settings. The mechanism of Cavallo [2006] coincides with Bailey's in those cases but is applicable to all decision scenarios, including important allocation domains to which Bailey's is not.

mal, be close to envy-free, and close to proportional for every agent in expectation. Next we will motivate our relaxation of a hard efficiency constraint by observing that no efficient mechanisms exist for canonical fair division settings, independent of fairness criteria. Finally we will demonstrate that the redistribution mechanism of [Bailey, 1997; Cavallo, 2006] performs exceedingly well on all three metrics in cake-cutting and assignment problems; this is in opposition to the simpler VCG mechanism, which, generally speaking, performs well on envy but not well with respect to welfare and proportionality.

1.3 Preliminaries

There is a set of agents $I = \{1, \ldots, n\}$ and a compact set of outcomes A (potentially infinite), where each $a \in A$ is an n-tuple $(a_1, a_2, \ldots, a_n)$ representing an allocation for each agent $i \in I$.There is a typespace Θ which represents the set of possible valuations for allocations. The joint typespace is Θ^n, and for any $\theta = (\theta_1, \ldots, \theta_n) \in \Theta^n$ and $a = (a_1, \ldots, a_n) \in A$, each agent i's *value* is $v_i(\theta_i, a_i)$. A mechanism is a tuple (f, T) where $f : \Theta^n \rightarrow A$ is a choice function and $T = (T_1, \ldots, T_n)$ defines a transfer function $T_i : \Theta^n \rightarrow \Re$ for each agent $i \in I$. In a mechanism agents report types, and then allocations and transfer payments are made according to f and T, respectively. We use notation $f_i(\theta)$ to denote a_i for the outcome a chosen by f given type profile θ (i.e., $f(\theta) = a = (f_1(\theta), \ldots, f_n(\theta)) = (a_1, \ldots, a_n)$ for some $a \in A$). We assume, for each $i \in I$, that i is self-interested and acts to maximize a *quasilinear* utility function u_i. Given mechanism (f, T), true joint type θ, and reported type $\hat{\theta}$, i then obtains utility: $v_i(\theta_i, f_i(\hat{\theta})) + T_i(\hat{\theta})$.We will specifically consider two classes of decision problems: cake-cutting and assignment.

Cake-cutting: There is a single infinitely divisible good to be allocated. The good may be heterogeneous, so values may depend not just on "how much" but also "which part" of the cake is received. Though our formal approach is completely general, in the evaluation section we will consider the following special classes of valuation functions:

- *Linear satiation*: value is homogeneous over all sections of the cake, and increases linearly with quantity, at slope determined by the agent's type, until plateauing at 1. If agent i with type θ_i receives $x\%$ of the cake, he obtains value: $v_i(\theta_i, x) = \min\{1, x\theta_i\}$. This captures different "satiation rates".
- *Exponential*: value is homogeneous over all sections of the cake; if allocated $x\%$ of the cake, an agent i with type θ_i obtains value $v_i(\theta_i, x) = 1 - e^{-x\theta_i}$.
- *Piecewise constant*: if K is the set of "kinds" of cake, each agent i's type has a component $\theta_{i,k}$ for every distinct kind $k \in K$. If, for each $k \in K$, agent i is allocated $x_k\%$ of the cake of kind k, he obtains value: $\sum_{k \in K} x_k \theta_{i,k}$.

Assignment: There are n agents and a heterogenous set of m items. Each agent's type determines a value for each item, and each agent can be allocated no more than one item. [3]

[3]Equivalently one can imagine that each agent's value for a bundle is restricted to equal the max of its values for any single item in the bundle, in which case an efficient allocation would not allocate multiple items to one agent.

2 Fairness metrics when utility is transferable

We generalize the either/or notions of efficiency, envy-freeness, and proportionality to "rates" that can be computed for any problem instance (defined by a joint type θ). Throughout the paper we assume a context of strategyproofness—we will only discuss the rates with respect to strategyproof mechanisms—so the measures are computed with respect to the truthful outcome.

Definition 1 (Welfare rate)**.** *The ratio of the aggregate social welfare to the agents including payments, to the social value of the efficient allocation without payments. I.e., for mechanism (f, T) and joint type $\theta \in \Theta^n$:*

$$\frac{\sum_{i \in I}(v_i(\theta_i, f_i(\theta)) + T_i(\theta))}{\sum_{i \in I} v_i(\theta_i, f_i^*(\theta))} \tag{1}$$

For a *no-deficit* mechanism (one in which aggregate payments never exceed 0), the welfare rate is bounded above by 1. A mechanism that achieves *full social welfare* is one with a welfare rate of 1 for all $\theta \in \Theta^n$.

We now generalize the notions of envy-freeness and proportionality to "envy rate" and "disproportionality rate" representing the average extent throughout the population to which, respectively, an agent prefers the outcome for another agent, and an agent fails to obtain a "fair share" $1/n$ fraction of the utility he could obtain as a dictator. Both measures range between 0 and 1. In the spirit of fairness, the measures give equal weight to each agent's envy or disproportionality, in the sense that, e.g., the disproportionality measure for an agent who obtains only $\frac{\epsilon}{n} < \frac{1}{n}$ of his maximum possible utility u is the same whether u is minuscule or enormous.

Definition 2 (Envy rate)**.** *Let u_{max} denote the utility an agent would have experienced if he received, maximizing over all agents j, j's allocation and j's payment. The envy rate equals, averaging over all agents, the difference between u_{max} and the agent's utility, divided by u_{max}. I.e., for mechanism (f, T) and joint type $\theta \in \Theta^n$:*

$$\frac{1}{n} \sum_{i \in I} \frac{\max_{j \in I}\{v_i(\theta_i, f_j(\theta)) + T_j(\theta)\} - \{v_i(\theta_i, f_i(\theta)) + T_i(\theta)\}}{\max_{j \in I}\{v_i(\theta_i, f_j(\theta)) + T_j(\theta)\}} \tag{2}$$

The envy rate never goes below 0 since each agent's actual allocation is included in the maximization. Envy-freeness is equivalent to the requirement that the envy rate be 0 for every problem instance.

Definition 3 (Disproportionality rate)**.** *Averaging over all agents, the maximum of 0 and $1/n$ minus the ratio of an agent's allocation value plus payment to the value the agent would experience from obtaining his optimal allocation and no payment, divided by $1/n$. I.e., for mechanism (f, T) and joint type $\theta \in \Theta^n$:*

$$\frac{1}{n} \sum_{i \in I} \max\left\{0,\ \left(\frac{1}{n} - \frac{v_i(\theta_i, f_i(\theta)) + T_i(\theta)}{\max\limits_{a \in A} v_i(\theta_i, a_i)}\right) \Big/ \frac{1}{n}\right\} \tag{3}$$

The disproportionality rate is fixed to never be below 0 for any agent so that it penalizes the failure to meet traditional proportionality but does not reward a mechanism for going "above and beyond" proportionality for some agents; this is in the spirit of fairness. Traditional proportionality[4] is equivalent to the requirement that the disproportionality rate be 0 for every problem instance.

3 On the impossibility of full social welfare

In this section we consider the question of whether, even disregarding envy and proportionality considerations, a worst-case welfare rate of 1 ("full social welfare") can be achieved. In a setting where subsidies are not available, this is equivalent to the question of whether implementing a dominant strategy efficient choice function with a mechanism that is strongly budget-balanced (0 revenue, 0 deficit) is possible. To answer the question we must specify something about the problem setting, i.e., the typespace. Green and Laffont [1979] showed that for unrestricted values settings, no mechanism achieves full social welfare in dominant strategies. In the case of multi-unit auctions,[5] we can also deduce that no strategyproof mechanism achieves full social welfare by the results of Guo and Conitzer [2007] and Moulin [2009]: they (independently) derived the mechanism for that setting that has the worst-case welfare rate when values are positive but otherwise unrestricted, and that rate is lower than 1.

In an extended version of this paper we complement those results with a proof technique that allows us to consider arbitrary restricted settings, and apply a sufficient condition for the non-existence of mechanisms that achieve full social welfare. This theorem and proofs are omitted here due to space constraints. The result establishes that in any anonymous,[6] dominant strategy efficient, and strongly budget-balanced mechanism, for any two possible types θ, θ' in the typespace, letting SW_k be the social welfare that results when k agents have type θ and $n-k$ agents have type θ', a specific linear combination of $SW_0, SW_1, \ldots, SW_n$ must equal 0. This is only a necessary condition for the possibility of full welfare and far from a sufficient one, yet alone it is an extremely restrictive condition and can be applied to very directly show that full social welfare is impossible in settings including assignment and cake-cutting, even with highly restricted values.

Theorem 1. *For the assignment problem with any number of goods, if the agent value spaces are symmetric, smoothly connected, and include values 0 and x for each item, for some $x > 0$, there exists no anonymous, dominant strategy efficient, and strongly budget-balanced mechanism.*

Theorem 2. *For cake-cutting, if the typespace is symmetric, smoothly connected and admits linear satiation values with types in the range $[0, n-1]$ (where n is the number of agents), there exists no anonymous, dominant strategy efficient, and strongly budget-balanced mechanism.*

[4]The more basic idea of extending proportionality to a transferable utility context is not new; see, e.g., [Cramton *et al.*, 1987].

[5]The multi-unit auction setting is different from the assignment problem in that the goods are identical and so the problem can be described as simply choosing "who to serve" with an item.

[6]Anonymity requires that the expected utility obtained by two agents with the same type is the same, which is natural in the spirit of fairness.

4 The redistribution mechanism

While full social welfare may be impossible, this of course does not preclude the existence of solutions that obtain very good social welfare, i.e., achieve a high welfare rate in expectation. The most well-known general social choice mechanism is VCG; but though VCG always achieves an outcome in dominant strategies that maximizes the sum of agent values, it requires that much of this value be transferred away from the group (high "revenue"). In fact, amongst all mechanisms that choose outcomes that maximize aggregate value, VCG requires the *maximum* transfer of that value outside of the group (see Theorem 2.10 of [Cavallo, 2008]).

In settings that are extremely lacking of structure, such as settings where each agent's value function over outcomes is completely unrestricted, no improvement over VCG is possible. However, in practically all allocation settings values have significant structure—for instance, in single-item allocation an agent obtains 0 value for any outcome in which he does not receive the item. Exploiting this structure to improve social welfare is the idea introduced, for restricted settings, by Bailey [1997], and for general settings, by Cavallo [2006].[7] The general *redistribution mechanism* (RM) proposed in [Cavallo, 2006] is as follows: implement VCG, then pay each agent i a quantity equal to $1/n$ times the minimum VCG revenue that would result independent of the agent's mode of participation. In the versions of the cake-cutting and assignment problems we examine here, the redistribution payment reduces to $1/n$ times the revenue that would result if the agent were not present.

To illustrate the mechanism, consider the 3-agent (i, j, k), 3-item (A, B, C) assignment problem depicted in Table 1, which one can think of as room assignment, rent division for the purpose of narrative.

	v_i	v_j	v_k
A	500	600	800
B	900	1000	900
C	600	900	600

Table 1: 3-agent, 3-item assignment problem example.

The optimal allocation is A to agent k, B to i, and C to j. Omitting the details of computation, under VCG i pays $\$100$, and neither j or k pay anything. Under RM i pays $\$66.67$, and j and k are each paid $\$33.33$. On this instance the welfare rate under VCG is $\frac{2500}{2600}$ and under RM it is 1. The envy and disproportionality rates for both mechanisms are 0 here. If this were a room assignment, rent division problem where the rent for the house is $\$1500$, starting with the equal-share payments of $\$500$ each to ensure no-deficit, under VCG agent i ends up paying $\$600$ and the other two agents

[7]Unlike Cavallo's proposal, Bailey's mechanism is not feasible for cake-cutting unless we assume the type "no value for any amount of cake" is included in the typespace.

pay $\$500$ each—the surplus $\$100$ must be transferred outside of the group (e.g., to a charity that no agent obtains utility from giving to). Under RM i pays $\$566.67$, and the other two agents each pay $\$466.67$. In this fortuitous example there is no surplus; in general there may be a surplus, but under RM it is never greater (and is typically far less) than under VCG.

We will see in the next section that in both cake-cutting and assignment, VCG does well with respect to minimizing envy, but very poorly with respect to welfare and, typically, proportionality. RM typically does well in all three metrics. Though in some cases VCG achieves a lower envy rate, it is always dominated by RM in terms of welfare and proportionality.

Theorem 3. *On any problem instance, in any domain, RM has a weakly higher welfare rate and weakly lower disproportionality rate than VCG.*

In the case of assignment with a single good, it is particularly easy to compare the traditional binary fairness properties of VCG (which reduces to a Vickrey auction) and RM. RM reduces to the following simple form: the high bidder is allocated the good and pays the second highest bid, and every agent is paid $1/n$ times the second highest bid amongst the other agents.

Theorem 4. *In any single-item allocation problem instance, RM yields an outcome that is envy-free and proportional for at least $n-2$ agents. VCG yields an outcome that is envy-free for all agents but proportional for a maximum of 1 agent that has non-zero value for the item.*

5 Evaluation

In this section we evaluate VCG and RM along the metrics of welfare, envy, and disproportionality rates introduced in Section 2. We do an average case analysis, measuring the *expected value* of each rate given a probability distribution over agent values.[8] In cake-cutting,[9] we examine values drawn from the linear satiation class (with typespace $[0, n]$), the exponential class (with typespace $[0, 9]$), and the piecewise constant class (with 3 kinds of cake[10] and value space $[0, 1]$ for each kind). The results are given in Table 2. We report results for a type distribution that is uniform over the typespace (we also considered Gaussian type distributions, but the results were very similar and are thus omitted); in the case of piecewise constant values the typespace is multidimensional, and we considered values that are uniformly distributed and independent across different kinds of cake. In all three cases VCG performs poorly with respect to welfare and proportionality, but has a low envy rate. RM performs well along all three measures, notably with welfare going to 1 and envy and disproportionality to 0 as the population size grows.

[8]Expected values were computed by a Monte Carlo sampling method, with each data point averaged over 2000–10000 (depending on the setting) randomly drawn joint type instances.

[9]When utilities are a concave function of quantity allocated (as we consider here), optimal allocations can be computed with a greedy algorithm that allocates each incremental crumb to the agent whose marginal utility per crumb is currently highest.

[10]Variants with more or less kinds (heterogeneity) of cake were considered; results were very similar.

metric	n	VCG	RM	VCG	RM	VCG	RM
WR	3	0.566	0.728	0.719	0.825	0.333	0.778
	5	0.505	0.852	0.569	0.898	0.200	0.920
	10	0.459	0.936	0.417	0.956	0.100	0.980
	15	0.442	0.959	0.347	0.974	0.067	0.991
ER	3	0.032	0.116	0.041	0.041	0	0.011
	5	0.029	0.076	0.021	0.012	0	0.011
	10	0.018	0.026	0.006	0.002	0	0.007
	15	0.015	0.013	0.003	0.001	0	0.004
DR	3	0.361	0.171	0.126	0.041	0.532	0.050
	5	0.376	0.027	0.224	0.000	0.693	0.002
	10	0.373	0.000	0.355	0.000	0.835	0.000
	15	0.375	0.000	0.431	0.000	0.887	0.000
		(a)		(b)		(c)	

Table 2: **Cake-cutting**. Expected welfare (WR), envy (ER), and disproportionality (DR) rates under VCG and RM in three cake-cutting settings: (a) homogeneous, with values that rise linearly in quantity with slope equal to the agent's type, until reaching 1; (b) homogeneous, with values that equal $1 - e^{-x\theta_i}$ for an agent with type θ_i that receives $x\%$ of the cake; and (c) heterogeneous, with values linear in quantity of each kind of cake, with distinct slope for each kind.

In the assignment problem, each agent's type is represented as a vector of m values, one for each item. In our evaluation we take values drawn independently and uniformly over $[0, 1]$ for each item. We examined the following cases, with n the number of agents: n items; $n - 1$ items; and $n - 2$ items. The results are depicted in Table 3. Somewhat surprisingly, in the classical linear assignment problem (n agents, n items; Table 3 (a)) we find that VCG is a serviceable solution, obtaining a reasonably high welfare rate, zero envy, and a low disproportionality rate. Moving to RM improves the welfare rate at the cost of a marginal increase in the envy rate. In the case of $n-1$ items (Table 3 (b)), neither VCG nor RM achieve near-optimal performance: although RM's welfare rate is significantly better than VCG's, both are poor. When there are $n - 2$ goods (Table 3 (c)), VCG is poor while RM shines.

Finally we consider the case of assignment with one good, i.e., single-item allocation. In this case alone, there is another strategyproof mechanism in the literature to which we can compare VCG and RM: the worst-case optimal mechanism proposed by Guo and Conitzer [2007] and Moulin [2009] (we'll call it GCM). The mechanism has no concise form, and is instead specified by a system of equations that depends on the number of agents, so we refer the reader to the source papers for its description. As illustrated in Table 4, both RM and GCM perform superbly with respect to welfare and proportionality; VCG's welfare and disproportionality rates are abysmal, but it achieves no-envy, as in all assignment problems. The differences in performance between RM and GCM on welfare and disproportionality are negligible, but RM's expected envy rate is only about $1/3$ of GCM's.

6 Conclusion

In many group decision-making settings approaches that excel at meeting welfare or fairness criteria, but not both, will be unsatisfactory; broader evaluation metrics and different so-

metric	n	VCG	RM	VCG	RM	VCG	RM
WR	3	0.882	0.907	0.457	0.528	0.337	0.781
	5	0.864	0.915	0.372	0.491	0.281	0.833
	10	0.878	0.94	0.269	0.389	0.2	0.901
	15	0.895	0.955	0.211	0.318	0.16	0.932
ER	3	0	0.02	0	0.233	0	0.195
	5	0	0.021	0	0.171	0	0.1
	10	0	0.013	0	0.109	0	0.044
	15	0	0.009	0	0.082	0	0.026
DR	3	0.015	0.013	0.463	0.391	0.765	0.202
	5	0.001	0.001	0.31	0.183	0.532	0.007
	10	0.000	0.000	0.162	0.041	0.301	0.000
	15	0.000	0.000	0.111	0.012	0.208	0.000
		(a)		(b)		(c)	

Table 3: **Assignment**. Welfare (WR), envy (ER), and disproportionality (DR) rates under VCG and RM in the assignment problem with n agents and different numbers of items: (a) n items; (b) $n-1$ items; and (c) $n-2$ items.

metric	n	VCG	RM	GCM
welfare	3	0.334	0.774	0.774
	5	0.196	0.921	0.893
	10	0.1	0.98	0.991
	15	0.067	0.991	~ 1.0
envy	3	0	0.199	0.199
	5	0	0.056	0.126
	10	0	0.012	0.037
	15	0	0.005	0.015
disproportionality	3	0.764	0.207	0.207
	5	0.867	0.057	0.069
	10	0.935	0.012	0.011
	15	0.957	0.005	0.005

Table 4: **Single-item**. Welfare, envy, and disproportionality rates under VCG, RM, and GCM in single-item assignment.

lutions are called for. When utility is quasilinear in money, mechanisms using payments can be considered, allowing us to elicit truthful participation, formulate meaningful measures of both welfare and fairness, and even "redistribute" utility. If agents are strategic it is impossible to achieve *full* social welfare (efficient allocation with no aggregate payments outside the group), but the redistribution mechanism—pre-existing in the literature—comes close in the canonical fair division settings, particularly for larger groups of agents. At the same time, the redistribution mechanism approximates the traditional fairness criteria of envy-freeness and proportionality. This makes it a compelling solution for division of goods when utility is transferable and the objective is fairness, welfare, or achieving both simultaneously.

References

[Abdulkadiroglu *et al.*, 2004] Atila Abdulkadiroglu, Tayfun Snmez, and M. Utku nver. Room assignment-rent division: A market approach. *Social Choice and Welfare*, 22(3):515–530, 2004.

[Alkan *et al.*, 1991] Ahmet Alkan, Gabrielle Demange, and David Gale. Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59(4):1023–1039, 1991.

[Bailey, 1997] Martin J. Bailey. The demand revealing process: To distribute the surplus. *Public Choice*, 91:107–126, 1997.

[Brams and Kilgour, 2001] Steven J. Brams and D. M. Kilgour. Competitive fair division. *Journal of Political Economy*, 109:418–443, 2001.

[Brams and Taylor, 1996] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, UK, 1996.

[Brams *et al.*, 2006] Steven J. Brams, Michael A. Jones, and Christian Klamler. Better ways to cut a cake. *Notices of the AMS*, 53(11):1314–1321, 2006.

[Cavallo, 2006] Ruggiero Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. In *Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 882–889, 2006.

[Cavallo, 2008] Ruggiero Cavallo. *Social Welfare Maximization in Dynamic Strategic Decision Problems*. Ph.D. Thesis, Harvard University, 2008.

[Chen *et al.*, 2010] Yiling Chen, John Lai, David C. Parkes, and Ariel D. Procaccia. Truth, justice, and cake cutting. In *Proceedings of the 24th Annual Conference on Artificial Intelligence (AAAI-10)*, pages 56–61, 2010.

[Clarke, 1971] Edward Clarke. Multipart pricing of public goods. *Public Choice*, 8:19–33, 1971.

[Cohen *et al.*, 2010] Edith Cohen, Michal Feldman, Amos Fiat, Haim Kaplan, and Svetlana Olonetsky. Truth and envy in capacitated allocation games. unpublished, 2010.

[Cramton *et al.*, 1987] P. Cramton, R. Gibbons, and P. Klemperer. Dissolving a partnership efficiently. *Econometrica*, 55(3):615–632, 1987.

[Green and Laffont, 1977] Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.

[Green and Laffont, 1979] Jerry R. Green and Jean-Jacques Laffont. *Incentives in public decision-making*. North Holland, New York, 1979.

[Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[Guo and Conitzer, 2007] Mingyu Guo and Vincent Conitzer. Worst-case optimal redistribution of VCG payments. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC-07), San Diego, CA, USA*, pages 30–39, 2007.

[Haake *et al.*, 2002] Claus-Jochen Haake, Matthias G. Raith, and Francis Edward Su. Bidding for envy-freeness: A procedural approach to n-player fair-division problems. *Social Choice and Welfare*, 19(4):723–749, 2002.

[Holmstrom, 1979] Bengt Holmstrom. Groves' scheme on restricted domains. *Econometrica*, 47(5):1137–1144, 1979.

[Hurwicz, 1960] Leonid Hurwicz. Optimality and informational efficiency in resource allocation processes. In Karlin Arrow and Suppes, editors, *Mathematical Methods in the Social Sciences*. Stanford University Press, 1960.

[Leonard, 1983] Herman B. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–479, 1983.

[Lipton *et al.*, 2004] Richard Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM conference on Electronic commerce (EC-04)*, 2004.

[Mossel and Tamuz, 2010] Elchanan Mossel and Omer Tamuz. Truthful fair division. In *Proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT-10)*, pages 288–299, 2010.

[Moulin, 2009] Hervé Moulin. Almost budget-balanced VCG mechanisms to assign multiple objects. *Journal of Economic Theory*, 144:96–119, 2009.

[Moulin, 2010] Hervé Moulin. Auctioning or assigning an object: some remarkable VCG mechanisms. *Social Choice and Welfare*, 34:193–216, 2010.

[Neyman, 1946] J. Neyman. Un théorème d'existence. *Centre Recherche Academie de Science Paris 222*, pages 843–845, 1946.

[Papai, 2003] S. Papai. Groves sealed bid auctions of heterogenous objects with fair prices. *Social Choice and Welfare*, 20(3):371–386, 2003.

[Porter *et al.*, 2004] R. Porter, Y. Shoham, and M. Tennenholtz. Fair imposition. *Journal of Economic Theory*, 118(2):209–228, Oct 2004.

[Steinhaus, 1948] H. Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[Vickrey, 1961] William Vickrey. Counterspeculations, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

Dominating Manipulations in Voting with Partial Information

Vincent Conitzer
Department of Computer Science
Duke University
Durham, NC 27708, USA
conitzer@cs.duke.edu

Toby Walsh
NICTA and UNSW
Sydney, Australia
toby.walsh@nicta.com.au

Lirong Xia
Department of Computer Science
Duke University
Durham, NC 27708, USA
lxia@cs.duke.edu

Abstract

We consider manipulation problems when the manipulator only has partial information about the votes of the non-manipulators. Such partial information is described by an *information set*, which is the set of profiles of the non-manipulators that are indistinguishable to the manipulator. Given such an information set, a *dominating manipulation* is a non-truthful vote that the manipulator can cast which makes the winner at least as preferable (and sometimes more preferable) as the winner when the manipulator votes truthfully. When the manipulator has full information, computing whether or not there exists a dominating manipulation is in P for many common voting rules (by known results). We show that when the manipulator has no information, there is no dominating manipulation for many common voting rules. When the manipulator's information is represented by partial orders and only a small portion of the preferences are unknown, computing a dominating manipulation is NP-hard for many common voting rules. Our results thus throw light on whether we can prevent strategic behavior by limiting information about the votes of other voters.

1 Introduction

In computational social choice, one appealing escape from the Gibbard-Satterthwaite theorem [12; 14] was proposed in [2]. Whilst manipulation may always be possible, perhaps it is computationally too difficult to find? Many results have subsequently been proven showing that various voting rules are NP-hard to manipulate [1; 5; 7; 6; 9; 17; 10] in various senses. However, recent results suggest that computing a manipulation is easy on average or in many cases. Therefore, computational complexity seems to be a weak barrier against manipulation. See [8; 11] for some surveys of this recent research.

It is normally assumed that the manipulator has full information about the votes of the non-manipulators. The argument often given is that if it is NP-hard with full information, then it only can be at least as computationally difficult with partial information. However, when there is only one manipulator, computing a manipulation is polynomial for most common voting rules, including all positional scoring rules, Copeland, maximin, and voting trees. The only known exceptions are STV [1] and ranked pairs [17]. Therefore, it is not clear whether a single manipulator has incentive to lie when the manipulator only has partial information.

In this paper, we study the problem of how one manipulator computes a manipulation based on partial information about the other votes. For example, the manipulator may know that some voters prefer one alternative to another, but might not be able to know all pairwise comparisons for all voters. We suppose the knowledge of the manipulator is described by an *information set* E. This is some subset of possible profiles of the non-manipulators which is known to contain the true profile. Given an information set and a pair of votes U and V, if for every profile in E, the manipulator is not worse off voting U than voting V, and there exists a profile in E such that the manipulator is strictly better off voting U, then we say that U *dominates* V. If there exists a vote U that dominates the true preferences of the manipulator then the manipulator has an incentive to vote untruthfully. We call this a *dominating manipulation.* If there is no such vote, then a risk-averse manipulator might have little incentive to vote strategically.

We are interested in whether a voting rule r is *immune* to dominating manipulations, meaning that a voter's true preferences are never dominated by another vote. If r is not immune to dominating manipulations, we are interested in whether r is *resistant*, meaning that computing whether a voter's true preferences are dominated by another vote U is NP-hard, or *vulnerable*, meaning that this problem is in P. These properties depend on both the voting rule and the form of the partial information. Interestingly, it is not hard to see that most voting rules are immune to manipulation when the partial information is just the current winner. For instance, with any majority consistent rule (for example, plurality), a risk averse manipulator will still want to vote for her most preferred alternative. This means that the chairman does not need to keep the current winner secret to prevent such manipulations. On the other hand, if the chairman lets slip more information, many rules stop being immune. With most scoring rules, if the manipulator knows the current scores, then the rule is no longer immune to such manipulation. For instance, when her most preferred alternative is too far behind to win, the manipulator might vote instead for a less preferred candidate who can win.

In this paper, we focus on the case where the partial information is represented by a profile P_{po} of partial orders, and the information set E consists of all linear orders that extend P_{po}. The dominating manipulation problem is related to the *possible/necessary winner* problems [13; 15; 4; 3; 16]. In possible/necessary winner problems, we are given an alternative c and a profile of partial orders P_{po} that represents the partial information of the voters' preferences. We are asked whether c is the winner for *some* extension of P_{po} (that is, c is a *possible winner*), or whether c is the winner for *every* extension of P_{po} (that is, c is a *necessary winner*). We note that in the possible/necessary winner problems, there is no manipulator and P_{po} represents the chair's partial information about the votes. In dominating manipulation problems, P_{po} represents the partial information of the manipulator about the non-manipulators.

We start with the special case where the manipulator has complete information. In this setting the dominating manipulation problem reduces to the standard manipulation problem, and many common voting rules are vulnerable to dominating manipulation (from known results). When the manipulator has no information, we show that a wide range of common voting rules are immune to dominating manipulation. When the manipulator's partial information is represented by partial orders, our results are summarized in Table 1.

	DOMINATING MANIPULATION[1]
STV	Resistant (Proposition 2)
Ranked pairs	Resistant (Proposition 2)
Borda	Resistant (Theorem 4)
Copeland	Resistant (Corollary 2)
Voting trees	Resistant (Corollary 2)
Maximin	Resistant (Theorem 7)
Plurality	Vulnerable (Algorithm 2)
Veto	Vulnerable (Omitted due to the space constraint.)

Table 1: Computational complexity of the dominating manipulation problems with partial orders, for common voting rules.

Our results are encouraging. For most voting rules r we study in this paper (except plurality and veto), hiding even a little information makes r resistant to dominating manipulation. If we hide all information, then r is immune to dominating manipulation. Therefore, limiting the information available to the manipulator appears to be a promising way to prevent strategic voting.

2 Preliminaries

Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be the set of *alternatives* (or *candidates*). A linear order on $\mathcal{C}$ is a transitive, antisymmetric, and total relation on $\mathcal{C}$. The set of all linear orders on $\mathcal{C}$ is denoted by $L(\mathcal{C})$. An n-voter profile P on $\mathcal{C}$ consists of n linear orders on $\mathcal{C}$. That is, $P = (V_1, \ldots, V_n)$, where for every $j \leq n$, $V_j \in L(\mathcal{C})$. The set of all n-profiles is denoted by $\mathcal{F}_n$. We let m denote the number of alternatives. For any linear order $V \in L(\mathcal{C})$ and any $i \leq m$, $\mathrm{Alt}(V, i)$ is the alternative that is ranked in the ith position in V. A *voting rule* r is a function that maps any profile on $\mathcal{C}$ to a unique winning alternative, that is, $r : \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \to \mathcal{C}$. The following are some common voting rules. In this paper, if not mentioned specifically, ties are broken in the fixed order $c_1 \succ c_2 \succ \cdots \succ c_m$.

• *(Positional) scoring rules*: Given a *scoring vector* $\vec{s}_m = (\vec{s}_m(1), \ldots, \vec{s}_m(m))$ of m integers, for any vote $V \in L(\mathcal{C})$ and any $c \in \mathcal{C}$, let $\vec{s}_m(V, c) = \vec{s}_m(j)$, where j is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $\vec{s}_m(P, c) = \sum_{j=1}^{n} \vec{s}_m(V_j, c)$. The rule will select $c \in \mathcal{C}$ so that $\vec{s}_m(P, c)$ is maximized. We assume scores are integers and decreasing. Some examples of positional scoring rules are *Borda*, for which the scoring vector is $(m-1, m-2, \ldots, 0)$, *plurality*, for which the scoring vector is $(1, 0, \ldots, 0)$, and *veto*, for which the scoring vector is $(1, \ldots, 1, 0)$.

• *Copeland*: For any two alternatives c_i and c_j, we conduct a *pairwise election* in which we count how many votes rank c_i ahead of c_j, and how many rank c_j ahead of c_i. c_i wins if and only if the majority of voters rank c_i ahead of c_j. An alternative receives one point for each such win in a pairwise election. Typically, an alternative also receives half a point for each pairwise tie, but this will not matter for our results. The winner is the alternative with the highest score.

• *Maximin*: Let $D_P(c_i, c_j)$ be the number of votes that rank c_i ahead of c_j minus the number of votes that rank c_j ahead of c_i in the profile P. The winner is the alternative c that maximizes $\min\{D_P(c, c') : c' \in \mathcal{C}, c' \neq c\}$.

• *Ranked pairs*: This rule first creates an entire ranking of all the alternatives. In each step, we will consider a pair of alternatives c_i, c_j that we have not previously considered; specifically, we choose the remaining pair with the highest $D_P(c_i, c_j)$. We then fix the order $c_i \succ c_j$, unless this contradicts previous orders that we fixed (that is, it violates transitivity). We continue until we have considered all pairs of alternatives (hence we have a full ranking). The alternative at the top of the ranking wins.

• *Voting trees*: A voting tree is a binary tree with m leaves, where each leaf is associated with an alternative. In each round, there is a pairwise election between an alternative c_i and its sibling c_j: if the majority of voters prefer c_i to c_j, then c_j is eliminated, and c_i is associated with the parent of these two nodes. The alternative that is associated with the root of the tree (i.e. wins all its rounds) is the winner.

• *Single transferable vote (STV)*: The election has m rounds. In each round, the alternative that gets the lowest plurality score (the number of times that the alternative is ranked in the top position) drops out, and is removed from all of the votes (so that votes for this alternative transfer to another alternative in the next round). The last-remaining alternative is the winner.

For any profile P, we let $\mathrm{WMG}(P)$ denote the *weighted majority graph* of P, defined as follows. $\mathrm{WMG}(P)$ is a directed graph whose vertices are the alternatives. For $i \neq j$, if $D_P(c_i, c_j) > 0$, then there is an edge (c_i, c_j) with weight $w_{ij} = D_P(c_i, c_j)$.

We say that a voting rule r is based on the *weighted majority graph (WMG)*, if for any pair of profiles P_1, P_2 such that $\mathrm{WMG}(P_1) = \mathrm{WMG}(P_2)$, we have $r(P_1) = r(P_2)$. A voting rule r is *Condorcet consistent* if it always selects the Condorcet winner (that is, the alternative that wins each of its

[1] All hardness results hold even when the number of undetermined pairs in each partial order is no more than a constant.

pairwise elections) whenever one exists.

3 Manipulation with Partial Information

We now introduce the framework of this paper. Suppose there are $n \geq 1$ non-manipulators and one manipulator. The information the manipulator has about the votes of the non-manipulators is represented by an *information set* E. The manipulator knows for sure that the profile of the non-manipulators is in E. However, the manipulator does not know exactly which profile in E it is. Usually E is represented in a compact way. Let $\mathcal{I}$ denote the set of all possible information sets in which the manipulator may find herself.

Example 1. *Suppose the voting rule is r.*

• *If the manipulator has no information, then the only information set is $E = \mathcal{F}_n$. Therefore $\mathcal{I} = \{\mathcal{F}_n\}$.*

• *If the manipulator has complete information, then $\mathcal{I} = \{\{P\} : P \in \mathcal{F}_n\}$.*

• *If the manipulator knows the current winner (before the manipulator votes), then the set of all information sets the manipulator might know is $\mathcal{I} = \{E_1, E_2, \ldots, E_m\}$, where for any $i \leq m$, $E_i = \{P \in \mathcal{F}_n : r(P) = c_i\}$.*

Let V_M denote the true preferences of the manipulator. Given a voting rule r and an information set E, we say that a vote U *dominates* another vote V, if for every profile $P \in E$, we have $r(P \cup \{U\}) \succeq_{V_M} r(P \cup \{V\})$, and there exists $P' \in E$ such that $r(P' \cup \{U\}) \succ_{V_M} r(P' \cup \{V\})$. In other words, when the manipulator only knows the voting rule r and the fact that the profile of the non-manipulators is in E (and no other information), voting U is a strategy that dominates voting V. We define the following two decision problems.

Definition 1. *Given a voting rule r, an information set E, the true preferences V_M of the manipulator, and two votes V and U, we are asked the following two questions.*

• *Does U dominate V? This is the* DOMINATION *problem.*

• *Does there exist a vote V' that dominates V_M? This is the* DOMINATING MANIPULATION *problem.*

We stress that usually E is represented in a compact way, otherwise the input size would already be exponentially large, which would trivialize the computational problems. Given a set $\mathcal{I}$ of information sets, we say a voting rule r is *immune* to dominating manipulation, if for every $E \in \mathcal{I}$ and every V_M that represents the manipulator's preferences, V_M is not dominated; r is *resistant* to dominating manipulation, if DOMINATING MANIPULATION is NP-hard (which means that r is not immune to dominating manipulation, assuming P≠NP); and r is *vulnerable* to dominating manipulation, if r is not immune to dominating manipulation, and DOMINATING MANIPULATION is in P.

4 Manipulation with Complete/No Information

In this section we focus on the following two special cases: (1) the manipulator has complete information, and (2) the manipulator has no information. It is not hard to see that when the manipulator has complete information, DOMINATING MANIPULATION coincides with the standard manipulation problem. Therefore, our framework of dominating manipulation is an extension of the traditional manipulation problem, and we immediately obtain the following proposition from the Gibbard-Satterthwaite theorem [12; 14].

Proposition 1. *When $m \geq 3$ and the manipulator has full information, a voting rule satisfies non-imposition and is immune to dominating manipulation if and only if it is a dictatorship.*

The following proposition directly follows from the computational complexity of the manipulation problems for some common voting rules [2; 1; 6; 18; 17].

Proposition 2. *When the manipulator has complete information, STV and ranked pairs are resistant to* DOMINATING MANIPULATION*; all positional scoring rules, Copeland, voting trees, and maximin are vulnerable to dominating manipulation.*

Next, we investigate the case where the manipulator has no information. We obtain the following positive results. Due to the space constraint, most proofs are omitted. All proofs are available on the third author's webpage.

Theorem 1. *When the manipulator has no information, any Condorcet consistent voting rule r is immune to dominating manipulation.*

Theorem 2. *When the manipulator has no information, Borda is immune to dominating manipulation.*

Theorem 3. *When the manipulator has no information and $n \geq 6(m-2)$, any positional scoring rule is immune to dominating manipulation.*

These results demonstrate that the information that the manipulator has about the votes of the non-manipulators plays an important role in determining strategic behavior. When the manipulator has complete information, many common voting rules are vulnerable to dominating manipulation, but if the manipulator has no information, then many common voting rules become immune to dominating manipulation.

5 Manipulation with Partial Orders

In this section, we study the case where the manipulator has partial information about the votes of the non-manipulators. We suppose the information is represented by a profile P_{po} composed of partial orders. That is, the information set is $E = \{P \in \mathcal{F}_n : P \text{ extends } P_{po}\}$. We note that the two cases discussed in the previous section (complete information and no information) are special cases of manipulation with partial orders. Consequently, by Proposition 1, when the manipulator's information is represented by partial orders and $m \geq 3$, no voting rule that satisfies non-imposition and non-dictatorship is immune to dominating manipulation. It also follows from Theorem 2 that STV and ranked pairs are resistant to dominating manipulation. The next theorem states that even when the manipulator only misses a tiny portion of the information, Borda becomes resistant to dominating manipulation.

Theorem 4. DOMINATION *and* DOMINATING MANIPULATION *with partial orders are* NP*-hard for Borda, even when the number of unknown pairs in each vote is no more than* 4*.*

Proof. We only prove that DOMINATION is NP-hard, via a reduction from EXACT COVER BY 3-SETS (X3C). The proof for DOMINATING MANIPULATION is omitted due to space

constraint. The reduction is similar to the proof of the NP-hardness of the possible winner problems under positional scoring rules in [16].

In an X3C instance, we are given two sets $\mathcal{V} = \{v_1, \ldots, v_q\}, \mathcal{S} = \{S_1, \ldots, S_t\}$, where for any $j \leq t, S_j \subseteq \mathcal{V}$ and $|S_j| = 3$. We are asked whether there exists a subset $\mathcal{S}'$ of $\mathcal{S}$ such that each element in $\mathcal{V}$ is in exactly one of the 3-sets in $\mathcal{S}'$. We construct a DOMINATION instance as follows.

Alternatives: $\mathcal{C} = \{c, w, d\} \cup \mathcal{V}$, where d is an auxiliary alternative. Therefore, $m = |\mathcal{C}| = q + 3$. Ties are broken in the following order: $c \succ w \succ \mathcal{V} \succ d$.

Manipulator's preferences and possible manipulation: $V_M = [w \succ c \succ d \succ \mathcal{V}]$. We are asked whether $V = V_M$ is dominated by $U = [w \succ d \succ c \succ \mathcal{V}]$.

The profile of partial orders: Let $P_{po} = P_1 \cup P_2$, defined as follows.

First part (P_1) of the profile: For each $j \leq t$, We define a partial order O_j as follows.

$O_j = [w \succ S_j \succ d \succ \text{Others}] \setminus [\{w\} \times (S_j \cup \{d\})]$

That is, O_j is a partial order that agrees with $w \succ S_j \succ d \succ$ Others, except that the pairwise relations between (w, S_j) and (w, d) are not determined (and these are the only 4 unknown relations). Let $P_1 = \{O_1, \ldots, O_t\}$.

Second part (P_2) of the profile: We first give the properties that we need P_2 to satisfy, then show how to construct P_2 in polynomial time. All votes in P_2 are linear orders that are used to adjust the score differences between alternatives. Let $P_1' = \{w \succ S_i \succ d \succ \text{Others} : i \leq t\}$. That is, P_1' ($|P_1'| = t$) is an extension of P_1 (in fact, P_1' is the set of linear orders that we started with to obtain P_1, before removing some of the pairwise relations). Let $\vec{s}_m = (m-1, \ldots, 0)$. P_2 is a set of linear orders such that the following holds for $Q = P_1' \cup P_2 \cup \{V\}$:

(1) For any $i \leq q$, $\vec{s}_m(Q, c) - \vec{s}_m(Q, v_i) = 1$, $\vec{s}_m(Q, w) - \vec{s}_m(Q, c) = 4q/3$.

(2) For any $i \leq q$, the scores of v_i and w, c are higher than the score of d in any extension of $P_1 \cup P_2 \cup \{V\}$ and in any extension of $P_1 \cup P_2 \cup \{U\}$.

(3) The size of P_2 is polynomial in $t + q$.

We now show how to construct P_2 in polynomial time. For any alternative $a \neq d$, we define the following two votes: $W_a = \{[a \succ d \succ \text{Others}], [\text{Rev}(\text{Others}) \succ a \succ d]\}$, where $\text{Rev}(\text{Others})$ is the reversed order of the alternatives in $\mathcal{C} \setminus \{a, d\}$. We note that for any alternative $a' \in \mathcal{C} \setminus \{a, d\}$, $\vec{s}_m(W, a) - \vec{s}_m(W, a') = 1$ and $\vec{s}_m(W, a') - \vec{s}_m(W, d) = 1$. Let $Q_1 = P_1' \cup \{V\}$. P_2 is composed of the following parts:

(1) $tm - \vec{s}_m(Q_1, c)$ copies of W_c.

(2) $tm + 4q/3 - \vec{s}_m(Q_1, w)$ copies of W_w.

(2) For each $i \leq q$, there are $tm - 1 - \vec{s}_m(Q_1, v_i)$ copies of W_{v_i}.

We next prove that V is dominated by U if and only if c is the winner in at least one extension of $P_{po} \cup \{V\}$. We note that for any $v \in \mathcal{V} \cup \{w\}$, the score of v in V is the same as the score of v in U. The score of c in U is lower than the score of c in V. Therefore, for any extension P^* of P_{po}, if $r(P^* \cup \{V\}) \in (\{w\} \cup \mathcal{V})$, then $r(P^* \cup \{V\}) = r(P^* \cup \{U\})$ (because d cannot win). Hence, for any extension P^* of P_{po}, voting U can result in a different outcome than voting V only if $r(P^* \cup V) = c$. If there exists an extension P^* of P_{po} such that $r(P^* \cup \{V\}) = c$, then we claim that the manipulator is strictly better off voting U than voting V. Let P_1^* denote the extension of P_1 in P^*. Then, because the total score of w is no more than the total score of c, w is ranked lower than d at least $\frac{q}{3}$ times in P_1^*. Meanwhile, for each $i \leq q$, v_i is not ranked higher than w more than one time in P_1^*, because otherwise the total score of v_i will be strictly higher than the total score of c. That is, the votes in P_1^* where $d \succ w$ make up a solution to the X3C instance. Therefore, the only possibility for c to win is for the scores of c, w, and all alternatives in $\mathcal{V}$ to be the same (so that c wins according to the tie-breaking mechanism). Now, we have $w = r(P^* \cup \{U\})$. Because $w \succ_{V_M} c$, the manipulator is better off voting U. It follows that V is dominated by U if and only if there exists an extension of $P_{po} \cup \{V\}$ where c is the winner.

The above reasoning also shows that V is dominated by U if and only if the X3C instance has a solution. Therefore, DOMINATION is NP-hard. □

Theorem 4 can be generalized to a class of scoring rules similar to the class of rules in Theorem 1 in [16], which does not include plurality or veto. In fact, as we will show later, plurality and veto are vulnerable to dominating manipulation.

We now investigate the relationship to the possible winner problem in more depth. In a possible winner problem (r, P_{po}, c), we are given a voting rule r, a profile P_{po} composed of n partial orders, and an alternative c. We are asked whether there exists an extension P of P_{po} such that $c = r(P)$. Intuitively, both DOMINATION and DOMINATING MANIPULATION seem to be harder than the possible winner problem under the same rule. Next, we present two theorems, which show that for any WMG-based rule, DOMINATION and DOMINATING MANIPULATION are harder than two special possible winner problems, respectively.

We first define a notion that will be used in defining the two special possible winner problems. For any instance of the possible winner problem (r, P_{po}, c), we define its *WMG partition* $\mathcal{R} = \{R_{c'} : c' \in \mathcal{C}\}$ as follows. For any $c' \in \mathcal{C}$, let $R_{c'} = \{\text{WMG}(P) : P \text{ extends } P_{po} \text{ and } r(P) = c'\}$. That is, $R_{c'}$ is composed of all WMGs of the extensions of P_{po}, where the winner is c'. It is possible that for some $c' \in \mathcal{C}$, $R_{c'}$ is empty. For any subset $\mathcal{C}' \subseteq \mathcal{C} \setminus \{c\}$, we let $G_{\mathcal{C}'}$ denote the weighted majority graph where for each $c' \in \mathcal{C}'$, there is an edge $c' \to c$ with weight 2, and these are the only edges in $G_{\mathcal{C}'}$. We are ready to define the two special possible winner problems for WMG-based voting rules.

Definition 2. *Let d^* be an alternative and let $\mathcal{C}'$ be a nonempty subset of $\mathcal{C} \setminus \{c, d^*\}$. For any WMG-based voting rule r, we let PW$_1(d^*, \mathcal{C}')$ denote the set of possible winner problems (r, P_{po}, c) satisfying the following conditions:*

1. *For any $G \in R_c$, $r(G + G_{\mathcal{C}'}) = d^*$.*
2. *For any $c' \neq c$ and any $G \in R_{c'}$, $r(G + G_{\mathcal{C}'}) = r(G)$.*
3. *For any $c' \in \mathcal{C}'$, $R_{c'} = \emptyset$.*

We recall that R_c and $R_{c'}$ are elements in the WMG partition of the possible winner problem.

Definition 3. *Let d^* be an alternative and let $\mathcal{C}'$ be a nonempty subset of $\mathcal{C} \setminus \{c, d^*\}$. For any WMG-based voting rule r, we let PW$_2(d^*, \mathcal{C}')$ denote the problem instances (r, P_{po}, c) of PW$_1(d^*, \mathcal{C}')$, where for any $c' \in \mathcal{C} \setminus \{c, d^*\}$, $R_{c'} = \emptyset$.*

Theorem 5. *Let r be a WMG-based voting rule. There is a polynomial time reduction from $\text{PW}_1(d^*, \mathcal{C}')$ to* DOMINATION *with partial orders, both under r.*

Proof. Let (r, P_{po}, c) be a $\text{PW}_1(d^*, \mathcal{C}')$ instance. We construct the following DOMINATION instance. Let the profile of partial orders be $Q_{po} = P_{po} \cup \{\text{Rev}(d^* \succ c \succ \mathcal{C}' \succ \text{Others})\}$, $V = V_M = [d^* \succ c \succ \mathcal{C}' \succ \text{Others}]$, and $U = [d^* \succ \mathcal{C}' \succ c \succ \text{Others}]$. Let P be an extension of P_{po}. It follows that $\text{WMG}(P \cup \{\text{Rev}(d^* \succ c \succ \mathcal{C}' \succ \text{Others}), V\}) = \text{WMG}(P)$, and $\text{WMG}(P \cup \{\text{Rev}(d^* \succ c \succ \mathcal{C}' \succ \text{Others}), U\}) = \text{WMG}(P) + G_{\mathcal{C}'}$. Therefore, the manipulator can change the winner if and only if $\text{WMG}(P) \in R_c$, which is equivalent to c being a possible winner. We recall that by the definition of $\text{PW}_1(d^*, \mathcal{C}')$, for any $G \in R_c$, $r(G + G_{\mathcal{C}'}) = d^*$; for any $c' \neq c$ and any $G \in R_{c'}$, $r(G + G_{\mathcal{C}'}) = c'$; and $d^* \succ_V c$. It follows that V (=V_M) is dominated by U if and only if the $\text{PW}_1(d^*, \mathcal{C}')$ instance has a solution. □

Theorem 5 can be used to prove that DOMINATION is NP-hard for Copeland, maximin, and voting trees, even when the number of undetermined pairs in each partial order is bounded above by a constant. It suffices to show that for each of these rules, there exist d^* and $\mathcal{C}'$ such that $\text{PW}_1(d^*, \mathcal{C}')$ is NP-hard. To prove this, we can modify the NP-completeness proofs of the possible winner problems for Copeland, maximin, and voting trees by Xia and Conitzer [16]. These proofs are omitted due to space constraint.

Corollary 1. DOMINATION *with partial orders is* NP*-hard for Copeland, maximin, and voting trees, even when the number of unknown pairs in each vote is bounded above by a constant.*

Theorem 6. *Let r be a WMG-based voting rule. There is a polynomial-time reduction from $\text{PW}_2(d^*, \mathcal{C}')$ to* DOMINATING MANIPULATION *with partial orders, both under r.*

Proof. The proof is similar to the proof for Theorem 5. We note that d^* is the manipulator's top-ranked alternative. Therefore, if c is not a possible winner, then V ($= V_M$) is not dominated by any other vote; if c is a possible winner, then V is dominated by $U = [w \succ \mathcal{C}' \succ c \succ \text{Others}]$. □

Similarly, we have the following corollary.

Corollary 2. DOMINATING MANIPULATION *with partial orders is* NP*-hard for Copeland and voting trees, even when the number of unknown pairs in each vote is bounded above by a constant.*

It is an open question if $\text{PW}_2(d^*, \mathcal{C}')$ with partial orders is NP-hard for maximin. However, we can directly prove that DOMINATING MANIPULATION is NP-hard for maximin by a reduction from X3C.

Theorem 7. DOMINATING MANIPULATION *with partial orders is* NP*-hard for maximin, even when the number of unknown pairs in each vote is no more than* 4*.*

For plurality and veto, there exist polynomial-time algorithms for both DOMINATION and DOMINATING MANIPULATION. Given an instance of DOMINATION, denoted by (r, P_{po}, V_M, V, U), we say that U is a *possible improvement* of V, if there exists an extension P of P_{po} such that $r(P \cup \{U\}) \succ_{V_M} r(P \cup \{V\})$. It follows that U dominates V if and only if U is a possible improvement of V, and V is not a possible improvement of U. We first introduce an algorithm (Algorithm 1) that checks whether U is a possible improvement of V for plurality.

Let c_{i^*} (resp., c_{j^*}) denote the top-ranked alternative in V (resp., U). We will check whether there exists $0 \leq l \leq n$, $d, d' \in \mathcal{C}$ with $d' \succ_{V_M} d$, and an extension P^* of P_{po}, such that if the manipulator votes for V, then the winner is d, whose plurality score in P^* is l, and if the manipulator votes for U, then the winner is d'. We note that if such d, d' exist, then either $d = c_{i^*}$ or $d' = c_{j^*}$ (or both hold). To this end, we solve multiple maximum-flow problems defined as follows.

Let $\mathcal{C}' \subset \mathcal{C}$ denote a set of alternatives. Let $\vec{e} = (e_1, \ldots, e_m) \in \mathbb{N}^m$ be an arbitrary vector composed of m natural numbers such that $\sum_{i=1}^{m} e_i \geq n$. We define a maximum-flow problem $F^{\vec{e}}_{\mathcal{C}'}$ as follows.

Vertices: $\{s, O_1, \ldots, O_n, c_1, \ldots, c_m, y, t\}$.

Edges:

- For any O_i, there is an edge from s to O_i with capacity 1.
- For any O_i and c_j, there is an edge $O_i \to c_j$ with capacity 1 if and only if c_j can be ranked in the top position in at least one extension of O_i.
- For any $c_i \in \mathcal{C}'$, there is an edge $c_i \to t$ with capacity e_i.
- For any $c_i \in \mathcal{C} \setminus \mathcal{C}'$, there is an edge $c_i \to y$ with capacity e_i.
- There is an edge $y \to t$ with capacity $n - \sum_{c_i \in \mathcal{C}'} e_i$.

For example, $F^{\vec{e}}_{\{c_1, c_2\}}$ is illustrated in Figure 1.

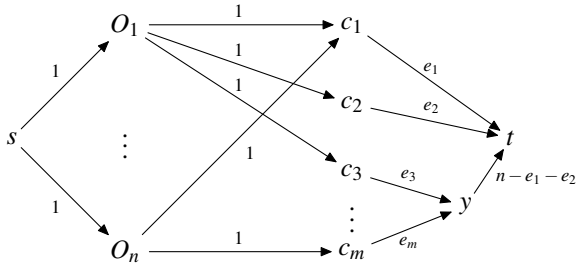


Figure 1: $F^{\vec{e}}_{\{c_1, c_2\}}$.

It is not hard to see that $F^{\vec{e}}_{\mathcal{C}'}$ has a solution whose value is n if and only if there exists an extension P^* of P_{po}, such that (1) for each $c_i \in \mathcal{C}'$, the plurality of c_i is exactly e_i, and (2) for each $c_{i'} \notin \mathcal{C}'$, the plurality of $c_{i'}$ is no more than $e_{i'}$. Now, for any pair of alternatives $d = c_i, d' = c_j$ such that $d' \succ_{V_M} d$ and either $d = c_{i^*}$ or $d' = c_{j^*}$, we define the set of *admissible maximum-flow problems* A^l_{Plu} to be the set of maximum flow problems $F^{\vec{e}}_{c_i, c_j}$ where $e_i = l$, and if $F^{\vec{e}}_{c_i, c_j}$ has a solution, then the manipulator can improve the winner by voting for U. Details are omitted due to space constraint. Algorithm 1 solves all maximum-flow problems in A^l_{Plu} to check whether U is a possible improvement of V.

The algorithm for DOMINATION (Algorithm 2) runs Algorithm 1 twice to check whether U is a possible improvement of V, and whether V is a possible improvement of U.

The algorithm for DOMINATING MANIPULATION for plurality simply runs Algorithm 2 $m - 1$ times. In the input

Algorithm 1: PossibleImprovement(V,U)

Let $c_{i^*} = \text{Alt}(V, 1)$ and $c_{j^*} = \text{Alt}(U, 1)$.
for *any* $0 \leq l \leq n$ *and any pair of alternatives* $d = c_i, d' = c_j$ *such that* $d' \succ_{V_M} d$ *and either* $d = c_{i^*}$ *or* $d' = c_{j^*}$ **do**
  Compute A^l_{Plu}.
  for *each maximum-flow problem* $F^{\vec{e}}_{\mathcal{C}'}$ *in* A^l_{Plu} **do**
    if $\sum_{c_i \in \mathcal{C}'} e_i \leq n$ *and the value of maximum flow in* $F^{\vec{e}}_{\mathcal{C}'}$ *is* n **then**
      Output that the U is a possible improvement of V, terminate the algorithm.
    end
  end
end
Output that U is not a possible improvement of V.

Algorithm 2: Domination

if *PossibleImprovement(V,U)="yes" and PossibleImprovement(U,V)="no"* **then**
  Output that V is dominated by U.
end
else
  Output that V is not dominated by U.
end

we always have that $V = V_M$, and for each alternative in $\mathcal{C} \setminus \{\text{Alt}(V, 1)\}$, we solve an instance where that alternative is ranked first in U. If in any step V is dominated by U, then there is a dominating manipulation; otherwise V is not dominated by any other vote. The algorithms for DOMINATION and DOMINATING MANIPULATION for veto are similar. We omit the details due to space constraint.

6 Future Work

Analysis of manipulation with partial information provides insight into what needs to be kept confidential in an election. For instance, in a plurality or veto election, revealing (perhaps unintentionally) part of the preferences of non-manipulators may open the door to strategic voting. An interesting open question is whether there are any more general relationships between the possible winner problem and the dominating manipulation problem with partial orders. It would be interesting to identify cases where voting rules are resistant or even immune to manipulation based on other types of partial information, for example, the set of possible winners. We may also consider other types of strategic behavior with partial information in our framework, for example, coalitional manipulation, bribery, and control. We are currently working on proving completeness results for higher levels of the polynomial hierarchy for problems similar to those studied in this paper.

Acknowledgments

Vincent Conitzer and Lirong Xia acknowledge NSF CAREER 0953756 and IIS-0812113, and an Alfred P. Sloan fellowship for support. Toby Walsh is supported by the Australian Department of Broadband, Communications and the Digital Economy, the ARC, and the Asian Office of Aerospace Research and Development (AOARD-104123). Lirong Xia is supported by a James B. Duke Fellowship. We thank all AAAI-11 and WSCAI reviewers for their helpful comments and suggestions.

References

[1] John Bartholdi, III and James Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[2] John Bartholdi, III, Craig Tovey, and Michael Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[3] Nadja Betzler and Britta Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *JCSS*, 76(8):812–836, 2010.

[4] Nadja Betzler, Susanne Hemmann, and Rolf Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. IJCAI*, pages 53–58, 2009.

[5] Vincent Conitzer and Tuomas Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proc. IJCAI*, pages 781–788, 2003.

[6] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007.

[7] Edith Elkind and Helger Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Proc. ISAAC*, 2005.

[8] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53:74–82, 2010.

[9] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Copeland voting: ties matter. In *Proc. AAMAS*, pages 983–990, 2008.

[10] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Manipulation of copeland elections. In *Proc. AAMAS*, pages 367–374, 2010.

[11] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[12] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41:587–602, 1973.

[13] Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. In *Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[14] Mark Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[15] Toby Walsh. Uncertainty in preference elicitation and aggregation. In *Proc. AAAI*, pages 3–8, 2007.

[16] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *JAIR*, pages 25–67, 2011.

[17] Lirong Xia, Michael Zuckerman, Ariel D. Procaccia, Vincent Conitzer, and Jeffrey Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proc. IJCAI*, pages 348–353, 2009.

[18] Michael Zuckerman, Ariel D. Procaccia, and Jeffrey S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412, 2009.

Randomised Room Assignment-Rent Division

Lachlan Dufton and **Kate Larson**
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
{ltdufton, klarson}@cs.uwaterloo.ca

Abstract

The room assignment-rent division problem allocates a heterogeneous set of indivisible items (e.g. rooms in a house) along with a share of some divisible item (e.g. the rent for the house), such that all items and resources are allocated without surplus or deficit, and each agent receives exactly one indivisible item. It is desirable to have envy-free outcomes but this is not possible for deterministic, truthful mechanisms. In this work we present truthful, randomised mechanisms for this problem, along with new measures of envy appropriate for non-deterministic mechanisms.

1 Introduction

The room assignment-rent division problem (RA-RD) [Su, 1999] is a classic problem in multiagent resource allocation and fair division. Consider a group of friends who will rent a house together. They must decide both who gets which room, and what share of the rent each person will pay. Each friend will want to be allocated just one room and there should be no surplus or deficit when meeting the total rent. Each individual has his or her own preferences on which room is best, such as preferring the largest room, or the room with the best view. More generally, this can be seen as a problem of allocating a set of indivisible, heterogeneous items (i.e. the rooms) along with a share of a divisible resource (i.e. the rent), such that all items are allocated and each agent gets exactly one indivisible item. The resources can have positive or negative utility.

While the real-estate setting is intuitive, this model of resource allocation can be applied to problems in other areas. In a job or task allocation setting, the indivisible items are tasks with some negative utility, and the divisible resource is some payment to be distributed among the workers upon completion of the tasks. The workers or processors can be considered the indivisible resources, with agents submitting work and covering some cost of maintaining the equipment.

In these settings, we are interested in more than just a Pareto-efficient allocation, but also some notion of fairness. In this paper we focus on *envy* and *envy-freeness* as measures of fairness. An allocation assigns a bundle to each agent, where a bundle is a single item along with some share of the divisible resource. For a particular allocation of bundles to agents, an agent is *envious* if it views another agent's bundle as strictly better than its own. An envy-free mechanism provides an allocation where no agent is envious. Brams and Taylor [1996] discuss envy-freeness and other measures of fairness in fair division.

In the RA-RD problem, deterministic, envy-free mechanisms are vulnerable to manipulation by the participating agents. Since envy-free allocations in this setting are Pareto-efficient with balanced agent transfers, this is a consequence of the impossibility result of Green and Laffont [1979]. As such, previous work on this problem has focussed on procedures that have full information about agent preferences.

We use randomisation to create new mechanisms that achieve envy-freeness in strategy-proof mechanisms. Randomisation has been a powerful technique for overcoming impossibility results in past work on social choice problems. For example, in other item allocation settings [Faltings, 2005], k-self-selection [Alon *et al.*, 2010], and voting protocols [Procaccia, 2010]. We also examine appropriate measures of qualities such as envy-freeness in randomised mechanisms. As previous work on the room assignment-rent division problem focusses on deterministic mechanisms, existing measures are not entirely suitable. For envy-freeness, we look at the probability a mechanism returns an envy-free outcome. Additionally we use the expected number of envy-free agents to consider what happens over all possible outcomes. We provide bounds for these measures in truthful mechanisms.

1.1 The Model

The room assignment-rent division problem assigns a set of indivisible, heterogeneous items (e.g. rooms in a shared house), M, to a set of agents, N such that all agents receive exactly one item and $|N| = |M|$. There is also some total quantity T of a divisible resource (e.g. rent) to be completely divided among the agents. This allocation and division is performed simultaneously. Each agent $i \in N$ has a value for each item $j \in M$, denoted as $v_i(j)$ (or equivalently, $v_{i,j}$), with the unit of the divisible resource as the numeraire.

We do not assume complete knowledge of agent types, so an RA-RD mechanism receives a vector of reported agent values $\bar{V} =< \bar{v}_1, \ldots \bar{v}_n >$ and produces an allocation function, $f : N \to M$, and a division $R \in \mathbb{R}^n$. A valid f must be bijective so every agent receives one item, every item is assigned to one agent. Let r_i denote the share of divisible re-

source agent i receives, where $\sum_{i \in N} r_i = T$. To simplify the notation, we let $v_i(f) = v_i(f(i))$. Agents have quasi-linear utilities, so an agent's utility for an allocation (f, R) is $u_i(f, R) = v_i(f) + r_i$.

In this work we use randomised mechanisms for the RA-RD problem. A deterministic mechanism takes a vector $\bar{V}$ of reported types and returns a single outcome, (f, R). A randomised mechanism instead uses a probability distribution over outcomes and returns a single outcome, (f, R), according to this distribution. Agents are risk neutral, so an agent's expected utility for a random distribution over outcomes is the probability-weighted sum of its utility of each outcome. A deterministic mechanism is truthful, or dominant strategy incentive compatible (DSIC), if no agent can increase its own utility by misreporting its type, regardless of other agents' actions. Similarly, a randomised mechanism is truthful *in expectation* if no agent can increase its own *expected* utility by misreporting, regardless of other agents' actions.

1.2 Related Work

For the room assignment-rent division problem there have been a number of previous solutions for finding envy-free solutions while assuming complete knowledge of agent types. Su [1999] proves the existence of envy-free outcomes for this setting, along with an interactive algorithm based on Sperner's lemma that uses simple queries to the agents. Abudlkadoiroğlu, Sönmez and Ünver [2004] developed an envy-free auction method for determining the allocation and prices of rooms with any number of agents that guarantees non-negative pricing. Haake, Raith and Su [2002] provided a more general procedure without the restrictions that the number of objects must equal the number of agents and each agent must receive exactly one object. For the room assignment-rent division problem, an envy-free solution relies on truthful preferences of the agents. Unfortunately, no deterministic mechanism exists that is both envy-free and non-manipulable.

Sun and Yang [2003] achieved a strategy-proof and envy-free mechanism for a similar allocation problem, but has different restrictions on the allocation of the divisible resource. Instead of dividing a single quantity of some resource, each indivisible item has its own "compensation limit". This model and proof was generalised by Andersson and Svensson [2008], and Andersson [2009] for greater flexibility on the indivisible objects, and a proof of coalitional strategy-proofness. However, the use of an item-based compensation limit instead of a single budget of divisible resource that must be entirely allocated mean that these mechanisms are incompatible for the room assignment-rent division model of this paper.

In this paper we use randomisation to achieve strategy-proof outcomes that are not possible in deterministic mechanisms. Moulin and Bogomolnaia [2001] and later Kojima [2009] examined a randomised mechanism for a similar allocation problem to RA-RD, but with the restriction that agents have the same ordinal ranking and where individual preferences are distinguished by a private "acceptance threshold". These randomised, strategy-proof mechanisms were shown to achieve efficient and envy-free outcomes. These papers also discuss methods of evaluating randomised allocation procedures. In a more general, but related item allocation setting, the Green-Laffont impossibility theorem [Green and Laffont, 1979] shows that for heterogeneous item allocation, no mechanism is Pareto-efficient, DSIC, and strong budget balanced. Strong budget balance requires that all agents' payments sum to exactly zero, while in RA-RD payments must sum to exactly T. Work by Faltings [2005] provided a randomised allocation technique that achieves incentive compatibility and budget balance at the expense of allocative efficiency. The quality of this randomised mechanism is assessed by the loss of efficiency in generated problems.

1.3 Deterministic RA-RD Mechanisms

As has been shown in previous work [Haake *et al.*, 2002; Su, 1999], no truthful, envy-free mechanism exists for the RA-RD problem. This follows by the Green-Laffont impossibility theorem [Green and Laffont, 1979], as an envy-free allocation is an efficient allocation [Alkan *et al.*, 1991], and the sum of payments must be budget balanced (if $T \neq 0$, each room can be given an initial charge of $\frac{T}{n}$ to bring the budget to zero). As the truthful mechanism cannot guarantee efficiency when ensuring the divisible resource is entirely allocated, the mechanism cannot provide an envy-free outcome for all inputs.

2 Envy-Freeness in Randomised Mechanisms

An envy-free, deterministic allocation mechanism produces an outcome where no agent prefers another agent's allocated bundle to its own. For the RA-RD problem, an outcome (f, R) is envy-free if :

$$v_i(f(i)) + r_i \geq v_i(f(j)) + r_j \ , \forall i, j \in N$$

This measurement states whether or not a single outcome is envy-free. When examining randomised mechanisms, which can produce several outcomes for a single input, this does not provide an appropriate comparison of mechanisms. For this problem, it is beneficial to consider measures of envy-freeness designed for randomised mechanisms. In a randomised mechanism, agent envy can be measured before the randomisation process (i.e. on the agent's lottery of outcomes), or on the final outcome. A simple extension of measuring envy to a randomised mechanism is to compare each agent's lottery of allocations, prior to the mechanism performing its random selection.

Definition 1. *For* ex ante envy-freeness*, no agent strictly prefers another agent's lottery over final outcomes. Let K be the set of allocations, and p_k is the probability of choosing $k \in K$, which has associated allocation and payment functions (f^k, R^k). That is, for all agents $i \in N$:*

$$\sum_{k \in K} p_k(v_i(f^k(i)) + r_i^k) \geq \sum_{k \in K} p_k(v_i(f^k(j)) + r_j^k) \ , \forall j \in N$$

Ex ante envy-freeness is trivial to achieve in truthful mechanisms – simply randomise over all possible allocations with equal probability and give each agent $\frac{T}{n}$ of the divisible resource. This gives the same lottery for each agent regardless of reported type but generally provides poor final outcomes, with most or all agents envious in all outcomes. Because of

this, we propose looking at envy-freeness in the actual outcomes, after the mechanism has performed the random selection. One measure is to look at which of these final outcomes are envy-free in the deterministic sense, and the probability of the mechanism producing such an outcome in the worst case.

Definition 2. *An outcome is envy-free if no agent values another agent's bundle higher than its own. The* guaranteed probability of envy-freeness *(GPEF) is the minimum probability a mechanism will produce an envy-free outcome, for any set of agents.*

The previous example that was *ex ante* envy-free has a GPEF of zero. For some sets of agents, it will never produce an envy-free outcome. Consider two agents that both prefer indivisible item 1. As the divisible resource is split evenly, whichever agent is assigned item 2 will be envious. This measure only considers the very best outcomes, where all agents are envy-free, and all other outcomes are assessed as valueless. For our third measure, we examine the level of envy, as the number of envious agents, in each of the possible outcomes.

Definition 3. *An envy-free agent is one who does not value another agent's bundle higher than its own in a particular allocation. The* expected number of envy-free agents *is the probability-weighted sum of the number of envy-free agents in each outcome of the mechanism for a particular input.*

In the example of two agents preferring the same item, the basic mechanism gives 1 expected envy-free agent, as both allocations would have one agent envious and one envy-free.

3 Randomised RA-RD Mechanisms

We now examine these new measures of envy-freeness in the RA-RD problem on mechanisms that are truthful *in expectation*. A mechanism is truthful in expectation if, irrespective of the actions of other agents, an agent's expected utility can not be increased by misreporting its type.

Lemma 1. *An RA-RD mechanism is truthful in expectation if (but not only if) each agent's* expected *share of the divisible resource, and probability of being assigned to each indivisible item is constant (independent of reported types).*

Proof. Let $p_{i,j}$ be the probability agent i is assigned item j, and $\bar{r_i} = E(r_i)$ be agent i's expected share of the divisible resource. The expected utility of agent $i \in N$ is calculated as: $E(u_i) = \sum_{j \in M} p_{i,j} v_i(j) + \bar{r_i}$. As all $p_{i,j}$ and $\bar{r_i}$ are constant with respect to the agent's bid/reported type, the agent's expected utility is constant and cannot be increased by misreporting. □

Note that these are not the necessary conditions for a truthful RA-RD mechanism. We use these conditions to define a simple, truthful mechanism as a baseline for comparing other randomised mechanisms.

A simple randomised RA-RD mechanism. From previous work [Alkan *et al.*, 1991; Haake *et al.*, 2002], given full knowledge of agents' types, we can find an envy-free allocation and division, denoted (f^*, R^*). Our random mechanism first calculates the envy-free solution, then randomly selects an integer $x \in [0, n-1]$. Agent i is given the item and share allocated to agent $(i + x) \pmod n$ in the envy-free allocation. Thus, $f^x(i) = f^*((i + x) \pmod n)$. Each agent has a $\frac{1}{n}$ probability of being assigned any particular item. An agent's expected share of the divisible resource is

$$\bar{r}_i = \sum_{j \in n} \frac{1}{n} r_j^* = \frac{1}{n} \sum_{j \in n} r_j^* = \frac{T}{n}$$

This is constant for each agent, so by Lemma 1 the mechanism is truthful in expectation, allowing the mechanism to correctly calculate (f^*, R^*).

Whenever $x = 0$, the envy-free outcome is chosen, and this occurs with probability $\frac{1}{n}$. Apart from special cases, for all other values of x, all agents will be envious of their bundle from the envy-free outcome. Thus, for this mechanism the GPEF is $\frac{1}{n}$. When $x = 0$, there are n envy-free agents, while in the worst case, all other choices of x will have no envy-free agents. This gives a worst-case expected number of envy-free agents of $n \cdot \frac{1}{n} + 0 \cdot \frac{n-1}{n} = 1$. In this mechanism, all agents have the same lottery over items and expected payment, so it is *ex ante* envy-free.

3.1 Maximising Guaranteed Probability of Envy-Freeness

A truthful mechanism that guarantees 100% probability of envy-freeness would be optimal for the three definitions in Section 2. Unfortunately, this is not possible for RA-RD.

Theorem 1. *A truthful (in expectation) mechanism for the RA-RD problem with n agents has a guaranteed probability of envy-freeness of* at most $\frac{1}{n}$.

Proof. In our setting with an equal number of agents and items, an envy-free allocation is a Pareto-efficient allocation [Alkan *et al.*, 1991]. So, if a mechanism were capable of envy-freeness with probability $p > \frac{1}{n}$, it would also provide an efficient allocation with probability at least p.

To get an efficient allocation with probability more than $\frac{1}{n}$, then all agents must be able to change their probabilities of item allocation through their reported values. For any mechanism for this problem, an agent's expected utility, which must be maximised when reporting truthfully, can be decomposed into parts. The first is its expected utility from receiving items – a probability-weighted sum of the resources it can receive. An agent will always receive one item. Next, the agent's *expected* payment for any mechanism can be separated into two functions $\bar{g}_i(v) + h_i(v_{-i})$. Function $\bar{g}_i(v)$ depends on all agents' reported types and must be maximised when agent i reports truthfully (similar to the Groves payment in a Vickrey-Clarke-Groves (VCG) mechanism). There is some additional expected payment, $h_i(v_{-i})$, that doesn't depend on agent i's reported type.

Let $p_{i,j}(v)$ denote the probability agent i receives item j. If an agent receives each item with equal probability, then the agent will receive some constant expected utility from the allocation, regardless of its reported type. The minimum probability an agent can receive an item, $\min_{i,v} p_{i,j}(v)$, determines the fraction of outcomes that contribute to this constant utility. All items must be received with probability *at least*

$\min_{i,v} p_{i,j}(v)$. So with n items, let $p_i^0 = n \cdot \min_{i,v} p_{i,j}(v)$ be the fraction of outcomes where each agent receives each item with equal probability. Reported values do not affect expected utility from these allocations and the contribution to $\bar{g}_i(v)$ to ensure truthfulness is 0.

If $p_i^*(v)$ is the probability an agent receives its item in the efficient allocation, then it receives this item with increased probability of $(p_i^*(v) - \frac{p_i^0}{n})$ over the equal-probability allocations that are independent of bids. For truthfulness, $\bar{g}_i(v)$ must include $(p_i^*(v) - \frac{p_i^0}{n})g_i(v)$, where $g_i(v) = \sum_{j \neq i} v_j(f^*(j))$ is the Groves payment. This maximises the agents expected utility when it bids such that the true efficient allocation is chosen. Finally, if $p_i^0(v) + p_i^*(v) < 1$, there are other, non-efficient allocations for which agent i can change the probability, but the mechanism cannot counteract any gain from misreporting without $\bar{g}_i(v)$ directly including agent i's reported values, which will allow agent i to benefit by reducing its payment. This gives an agent's final expected payment of $(1-p_i^0)g_i(v)+h_i(v_{-i})$. The efficient allocation is possible with probability at most $p_i^*(v) = (1 - p_i^0)$.

All agents' expected payments must sum to T. If $p_i^0 < 1$, then dividing the h functions by constant factor $(1 - p_i^0)$ would give budget balanced Groves transfers. This contradicts the Green-Laffont impossibility theorem, so $p_i^0 = 1$.

With constant probability of being assigned each item, an agent cannot change its expected utility from the allocation by misreporting. This limits the probability of an efficient allocation to at most $\frac{1}{n}$ in the worst case (where there is only a single efficient allocation). Envy-free outcomes have efficient allocations so the best GPEF is $\frac{1}{n}$. □

This is a tight bound as demonstrated by the simple randomised RA-RD mechanism described above, with a GPEF of $\frac{1}{n}$. This places some limiting restrictions on what is possible with a strategy-proof mechanism for this problem. Envy-freeness at a low probability that asymptotically goes to zero means that most of the time the mechanism will produce a "bad" result. Considering only envy-free outcomes ignores what happens in the remainder of cases. In the mechanism described above, in the $(n-1)$ non-envy-free outcomes, every single agent will be envious. This motivates measuring the quality of each outcome with more detail than a yes/no test of "envy-free".

3.2 Maximising Expected Number of Envy-Free Agents

While having all agents envy-free is the ideal outcome, attempting to maximise the probability of such an outcome can come at the expense of the quality of non-envy-free outcomes. For truthful mechanisms, these non-envy-free outcomes are the most likely, so when comparing mechanisms they should not be ignored.

The above mechanism with a GPEF of $\frac{1}{n}$ has expected number of envy-free agents of 1, as defined in Definition 3. This is because there is a $\frac{1}{n}$ probability of n envy-free agents, and 0 envy-free agents otherwise. By this measure alone, this is equivalent to a mechanism that always has 1 envy-free agent, such as a "random dictator" mechanism. The "random dictator" picks an agent at random and gives that agent its most preferred item along with the maximum share of the divisible resource (i.e. $\max(T, 0)$), with the remaining resources allocated to other agents independently of all agent bids. As the probability of being the dictator does not depend on reported types, no agent can benefit by misreporting its type.

The maximum expected number of envy-free agents is n, and this implies that every outcome is envy-free. However, as shown in the previous subsection, this is not possible for a truthful mechanism.

Theorem 2. *A truthful (in expectation) mechanism for the RA-RD problem with n agents has an expected number of envy-freeness of at most $(n - 1 + \frac{1}{n})$.*

Proof. From Theorem 1, the maximum probability of an envy-free outcome is $\frac{1}{n}$, where there are n envy-free agents. The remaining outcomes, with probability $\frac{n-1}{n}$, can have at most $(n-1)$ envy-free agents. This gives an expected number of envy-free agents of $n\frac{1}{n} + (n-1)\frac{n-1}{n} = n - 1 + \frac{1}{n}$ □

The GPEF was maximised with a fairly simple mechanism, and in the rest of this section we present mechanisms for maximising the expected number of envy-free agents. The first is a mechanism that achieves the bound in Theorem 2 for two agents, followed by a more general mechanism with expected number of envy-free agents of at least $(n - 1)$, falling short of the bound by $\frac{1}{n}$.

The 2 Agent Case

For $n = 2$, this bound, $\frac{3}{2}$, can be reached with the following mechanism. Let I_j denote the point of indifference for agent j, which is the division of the divisible resource such that all bundles have equal value. For two agents, this can be represented as a single value, as the divisions must sum to T, and can be calculated as:

$$v_{j,1} + I_j = v_{j,2} + (T - I_j) \Rightarrow \quad I_j = \frac{1}{2}(v_{j,2} - v_{j,1} + T)$$

The mechanism chooses an agent at random, and uses that agent's point of indifference to determine bundles. Agents are then randomly assigned to a bundle. Each agent has a $\frac{1}{2}$ probability of being assigned each indivisible resource, and has an constant expected share of the divisible resource:

$$\bar{r}_1 = \bar{r}_2 = \frac{1}{2}\left(\frac{I_1 + (T - I_1)}{2} + \frac{I_2 + (T - I_2)}{2}\right) = \frac{T}{2}$$

So by Lemma 1, this mechanism is truthful in expectation. The agent chosen to set the bundles will be envy-free with either bundle, while the other agent will prefer one bundle, so there is a probability of $\frac{1}{2}$ this agent will be envious. This gives expected number of envy-free agents of $\frac{3}{2}$ and a GPEF of $\frac{1}{2}$. Thus, based on both measures of envy-freeness, the worst-case behaviour cannot be improved.

The $n > 2$ Agent Case

Our mechanism is a random distribution over deterministic mechanisms that are modifications to a VCG allocation with $(n - 1)$ agents, based on the randomised technique proposed by Faltings [2005]. The mechanism proceeds as follows:

1. Find f, the efficient allocation for all agents in N. The value of this efficient allocation is $\bar{C} = \sum_{i \in N} v_i(f)$.
2. Next, randomly select an agent $x \in N$, with equal probability over all agents, as the agent to be "ignored".
3. Find f_{-x} and $f_{-\{i,x\}}$, the efficient allocations for agents $N \setminus \{x\}$ and $N \setminus \{i, x\}$ respectively, for all agents $i \neq x$.
4. Assign non-ignored agents according to f_{-x}, giving agent x the leftover item.
5. Agents make payments according to r_i^x for each agent $i \neq x$, and r_x^x for agent x, as in the following equations.

$$r_i^x = -C^x + v_i(f_{-x}(i)) + C_{-i}^x + \frac{T}{n} - \frac{\bar{C}}{n}, i \neq x \quad (1)$$

$$\begin{aligned} r_x^x &= T - \sum_{i \neq x} r_i^x \\ &= (n-2)C^x - \sum_{i \neq x} C_{-i}^x + \frac{T}{n} + \frac{(n-1)}{n}\bar{C} \end{aligned} \quad (2)$$

Where $C^x = \sum_{j \neq x} v_j(f_{-x}(j))$ is the value of the efficient allocation excluding x, and $C_{-i}^x = \sum_{j \neq \{i,x\}} v_j(f_{-\{i,x\}}(j))$ is the value of the efficient allocation excluding $\{x, i\}$.

The payment for agent x is calculated based on the other agents' payments to ensure strong budget balance, i.e. the sum of all payments is equal to T. The payment r_i^x is made up of three parts. The first three terms in Equation 1 are the VCG payments with Clarke pivot payments in an allocation setting with agent x ignored. For this part of the payment function, along with the allocation function f_{-x}, the agents will have no incentive to misreport. Additionally, VCG mechanisms with Clake pivot payments are known to be envy-free when agents only receive one item [Leonard, 1983; Cohen *et al.*, 2010], so there will be no envy between non-ignored agents. The term $\frac{T}{n}$ is added equally to all agents, so will not affect envy or truthfulness. It is added to ensure payments sum to T. The final term, $\frac{-\bar{C}}{n}$, is added to ensure no agents are envious of the ignored agent. It is added equally to all agents, so will not create envy between non-ignored agents. This breaks the incentive-compatibility of the VCG payments, as it depends on all agents' reported values. When considering *expected* utility, agents have a $\frac{1}{n}$ probability of paying $\frac{(n-1)}{n}\bar{C}$ and an $\frac{(n-1)}{n}$ probability of paying $\frac{-\bar{C}}{n}$, so in expected utility the term cancels out. This means the mechanism remains truthful in expectation. If the value of the efficient allocation is at least T, then all agents will have a non-negative expected utility.

While non-ignored agents are not envious of each other, the pricing must also ensure they are not envious of the ignored agent. Agent i is envious of agent x iff:

$$\begin{aligned} & v_i(f_{-x}(i)) - r_i^x < v_i(f_{-x}(x)) - r_x^x \\ \Rightarrow & \bar{C} < v_i(f_{-x}(x)) + C_{-i}^x + \sum_{j \neq x} C_{-j}^x - (n-1)C^x \end{aligned} \quad (3)$$

Since, assuming non-negative agent values, $C^x \geq C_{-i}^x$, then $\bar{C} \geq \bar{C} + \sum_{i \neq x} C_{-i}^x - (n-1)C^x$. Also, for any agents $\{i, x\}$, we have $\bar{C} \geq C^x \geq C_{-i}^x + v_i(f_{-x}(x))$. Otherwise the efficient allocation used for C^x could have been improved by using allocation $f_{-\{i,x\}}$ and switching agent i to item $f_{-x}(x)$. Thus we have:

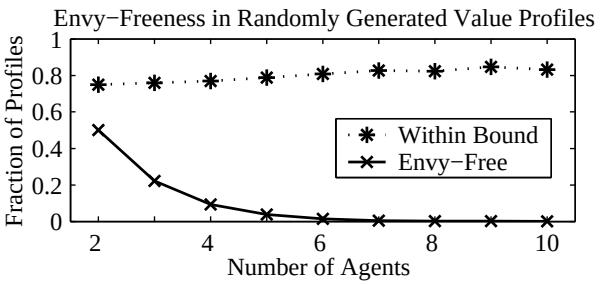


Figure 1: Fraction of value profiles that give outcomes within the worst case bounds, and where all outcomes are envy-free.

$$\bar{C} \geq v_i(f_{-x}(x)) + C_{-i}^x + \sum_{i \neq x} C_{-i}^x - (n-1)C^x$$

As no agent can be envious of the ignored agent, for any choice of x, there will be at least $(n-1)$ envy-free agents. This is the minimum expected number of envy-free agents, but short of the upper bound by $\frac{1}{n}$.

3.3 Empirical tests

The mechanism described in Section 3.2 for $n > 2$ agents does not meet the bound for guaranteed probability of envy-freeness or expected number of envy-free agents. While $(n-1)$ agents are guaranteed to be envy-free, the excluded agent may be envious in all outcomes. We test our mechanism empirically by generating random value profiles, where each agent's value for an item is drawn from a uniform distribution in the range $[0, 1]$. Including negative values did not noticeably affect our results. We then calculated the expected number of envy-free agents and the probability of envy-freeness for each value profile. At least 2500 random value profiles were generated for each n.

The plot in Figure 1 summarises the fraction of value profiles that give at least $\frac{1}{n}$ probability of envy-freeness, and the profiles that always give envy-freeness. For this mechanism, outcomes either have 0 or 1 envious agents, so all outcomes that give a probability of envy-freeness of at least $\frac{1}{n}$ also have an expected number of envy-free agents of at least $(n-1+\frac{1}{n})$. The dotted line in the plot shows that the majority of profiles fall within the optimal bound for these two measures, and this fraction increases with additional agents. However, there is still a significant fraction of profiles for which this mechanism falls short of this bound. So these worst-case profiles are not rare, special cases. The solid line shows that the fraction of ideal cases, where all outcomes are envy-free, for this mechanism rapidly approaches zero. So with this mechanism, an input that will always give an envy-free outcome becomes extremely rare as n increases.

4 Relation to Heterogeneous Item Allocation

This randomised approach to the room assignment-rent division problem, along with the measures used to assess randomised mechanisms can be used in related problems. The problem of budget balanced, efficient allocation involves distributing a set of heterogeneous items to a set of agents such

that the items are allocated efficiently, the sum of all agents' payments is zero (strong budget balance), and no agent benefits from misreporting preferences. In variations of this problem, there can be a different number of agents and items, and agents may not necessarily have unit demand. However, due to the Green-Laffont impossibility theorem, there is no efficient mechanism that is DSIC and strong budget balanced.

The RA-RD mechanism for $n > 2$ agents, described in Section 3.2, with the $\bar{C}$ and T terms removed from payment functions is strong budget balanced and DSIC. This is because the VCG mechanism used after an agent is ignored is DSIC and the ignored agent is payed so as to achieve strong budget balance. While not efficient deterministically, the Pareto efficiency of randomised mechanisms can be assessed by measures similar to those used for envy-freeness in RA-RD. For each choice of ignored agent, the remaining $(n-1)$ agents are assigned to an efficient allocation for those agents. Thus, in every outcome, the expected number of agents over which the allocation is efficient is at least $(n-1)$. This is similar to the property of expected number of envy-free agents. Note that this will hold for general allocation settings, not just those where each agent receives at most one item.

In the restricted case where each agent receives at most one item, and where $m \leq n$, for at least one chosen ignored agent the overall allocation for all n agents is efficient. For $n = m$, there is at least one agent who, when ignored, does not change the efficient allocation of the remaining agents. Furthermore, if $m < n$, then ignoring any of the agents that were left unallocated in the efficient allocation will also leave the allocation unchanged. In cases where the allocation is unchanged, then the final outcome will be efficient over all agents. As there are n different outcomes, and $n - m$ agents who receive no item in the efficient allocation, this gives a worst-case probability of an efficient allocation of $\frac{1}{n}$ for cases where $m = n$, or $\frac{n-m}{n}$ for cases where $m < n$. This measurement is analogous to the guaranteed probability of envy-freeness, and from Theorem 1 it is also the best achievable for $n = m$.

5 Conclusions and Future Work

In this work we presented randomised mechanisms for achieving envy-freeness in the room assignment-rent division problem. A deterministic mechanism is unable to provide an envy-free outcome while ensuring agents have no incentive to misreport their preferences. For a randomised mechanism, there are several possible outcomes, so evaluating and comparing these mechanisms by purely deterministic measures is not always suitable. We presented measures of envy-freeness appropriate for comparing randomised mechanisms.

Calculating envy between agents' lotteries of outcomes is not an effective measure in the RA-RD problem, as we show it is trivial to achieve this in mechanisms, and it does not consider the quality of final outcomes. Instead we focused on measuring the GPEF, which shows, in the worst case, what probability the mechanism will achieve the ideal outcome of envy-freeness in all agents. We also propose assessing mechanisms based on the expected number of envy-free agents, which can give an expected level of quality where the ideal outcome is unlikely. For these measures on the RA-RD problem, we provided upper bounds for strategy-proof randomised mechanisms.

These measures can be applied to mechanisms in other problems where truthful, deterministic, envy-free mechanisms are impossible. Similar measures can also be used on other qualities, such as Pareto efficiency. Efficiency cannot be achieved with strong budget balance and incentive compatibility, but a randomised mechanism can guarantee a minimum probability of efficiency in the worst-case.

References

[Abdulkadiroğlu *et al.*, 2004] A. Abdulkadiroğlu, T. Sönmez, and M. U. Ünver. Room assignment-rent division: A market approach. *Social Choice and Welfare*, 22(3):515–538, 2004.

[Alkan *et al.*, 1991] A. Alkan, G. Demange, and D. Gale. Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59(4):1023–39, 1991.

[Alon *et al.*, 2010] N. Alon, F. Fischer, A. D. Procaccia, and M. Tennenholtz. Sum of us: Strategyproof selection from the selectors. Working paper, March 2010.

[Andersson and Svensson, 2008] T. Andersson and L.-G. Svensson. Non-manipulable assignment of individuals to positions revisited. *Mathematical Social Sciences*, 56(3):350–354, 2008.

[Andersson, 2009] T. Andersson. A general strategy-proof fair allocation mechanism revisited. *Economics Bulletin*, 29(3):1717–1722, 2009.

[Brams and Taylor, 1996] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[Cohen *et al.*, 2010] E. Cohen, M. Feldman, A. Fiat, H. Kaplan, and S. Olonetsky. Truth and envy in capacitated allocation games. Discussion Paper Series, Hebrew University, Jerusalem, 2010.

[Faltings, 2005] B. Faltings. A budget-balanced, incentive-compatible scheme for social choice. In *Agent-Mediated Electronic Commerce VI*, 30–43., 2005.

[Green and Laffont, 1979] J. R. Green and J.-J. Laffont. *Incentives in public decision-making*. North-Holland Publishing Company, Amsterdam, 1979.

[Haake *et al.*, 2002] C.-J. Haake, M. G. Raith, and F. E. Su. Bidding for envy-freeness: A procedural approach to n-player fair-division problems. *Social Choice and Welfare*, 19(4):723–749, 2002.

[Kojima, 2009] F. Kojima. Random assignment of multiple indivisible objects. *Mathematical Social Sciences*, 57(1):134–142, 2009.

[Leonard, 1983] H. B. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–79, 1983.

[Moulin and Bogomolnaia, 2001] H. Moulin and A. Bogomolnaia. A simple random assignment problem with a unique solution. *Economic Theory*, 19(3):623–636, 2001.

[Procaccia, 2010] A. D. Procaccia. Can approximation circumvent Gibbard-Satterthwaite? In *Proc. of AAAI'10*, Atlanta, GA, 2010.

[Su, 1999] F. E. Su. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly*, 106(10):930–942, 1999.

[Sun and Yang, 2003] N. Sun and Z. Yang. A general strategy proof fair allocation mechanism. *Economics Letters*, 81(1):73–79, 2003.

Reassignment-Based Strategy-Proof Mechanism for Interdependent Task Allocation with Private Costs and Execution Failures

Ayman Ghoneim[1,2,*] **and Jussi Rintanen**[1]
The Australian National University[1] and NICTA[2†]
Canberra ACT, Australia

Abstract

In this study, we consider a task allocation model with interdependent tasks, where tasks are assigned based on what agents report about their privately known capabilities and costs. Since selfish agents may strategically misreport their private information in order to increase their payments, mechanism design is used to determine a payment schema that guarantees truthful reporting. Misreported information may cause execution failures, creating interdependencies between the agents' valuations. For this problem, efficient and strategy-proof mechanisms have not been proposed yet. In this study, we show that such mechanisms exist if the failing tasks are reassigned, in addition, individual rationality and center rationality are obtained. Then, we extend the model to consider agents who have limited resources, and show that the center rationality property is lost.

1 Introduction

Task allocation is an important and challenging problem that occurs in various real-life applications, ranging from construction, service providing, to computing and research projects. Adopting a general model, a center wants to assign some tasks to a number of self-interested agents, where each agent has its own private information (i.e., type) that describes its abilities and costs for executing tasks. Given that the center aims for an efficient assignment (i.e., one that maximizes the social welfare) and provides payments to the agents, agents may strategically misreport their types in order to increase their payments. Thus, mechanism design is used to determine the payments that guarantee truthful reporting.

In this study, we consider the interdependent task allocation (ITA) problem, where tasks may fail during the execution because of the agents' strategically misreported information (i.e., agents claim the ability to perform tasks that they cannot perform). This model of failures is suitable when assuming selfish agents, and for mimicking the one-shot interaction situations in which agents don't care much about future implications (e.g., reputation, future opportunities). Given the interdependencies between tasks, an agent may not be able to execute its assigned tasks if their predecessor tasks have failed. This implies that an agent's *actual value* of its assigned tasks may depend on other agents' *actual types*, and that agents in such settings have *interdependent valuations*. When valuations are interdependent, mechanisms that achieve the strongest and most preferable form of truthfulness in dominant strategy (i.e., strategy-proof) have *not* been proposed yet for *any* domain (see Section 5).

In this study, we *prove* that it is impossible for an efficient mechanism to achieve strategy-proofness using a single allocation round, even if agents have sufficient resources. Then, we *contribute* a novel efficient mechanism that achieves strategy-proofness by using multiple allocation rounds (i.e., reassign the failing tasks). Finally, we extend the ITA model to consider agents with limited resources, and *prove* that the center rationality property is lost. In the next section, we formulate the task allocation problem as a mechanism design problem. In Sections 3 and 4, we propose the reassignment mechanism and discuss limited resources. Section 5 discusses related work, and finally, we conclude the study and discuss future work in Section 6.

2 Task Allocation and Preliminary Concepts

Basic Model. Assume a center that has a set $T = \{t_1, \ldots, t_m\}$ of m tasks. There are *predefined* interdependencies (i.e., an ordering) between these tasks, where some tasks can't be executed unless their predecessor tasks were executed successfully. Thus, each task t may have a set of successor tasks $t_{\succ}$ and a set of predecessor tasks $t_{\prec}$. The center gains a reward $R(t)$ (e.g., a market value) for each successful task t. The center wants to allocate the tasks to a set α of n self-interested agents, where each agent has its own private information (i.e., type) and knows nothing about other agents' types. The type $\theta_i = \langle T_i; C_i(t), \forall t \in T_i \rangle$ of agent i consists of: *1.* the set of tasks $T_i \subseteq T$ that the agent can perform, and *2.* the cost $C_i(t)$ for which the agent can execute each task $t \in T_i$.

Outcome. The center wants to determine an assignment

*Author to whom correspondence should be addressed. Email: ayman.ghoneim@anu.edu.au

†NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

(i.e., outcome) $o = \{(t_1, i), (t_2, j), \ldots\}$, where each pair indicates the agent who is assigned a certain task, e.g., (t_1, i) means that agent i is assigned t_1. Under an outcome o, agent i is assigned the tasks in $T_i(o) = \{t_k | (t_k, i) \in o\}$, and $T_A(o) = \bigcup_{i \in \alpha} T_i(o)$ is the set of assigned tasks. An assignment may *not* contain all the offered tasks by the center in T, i.e., a task t and its successor tasks $t_{\succ}$ will not be assigned if no agent reported the ability to perform task t.

Tentative Values and Efficiency. The tentative value of agent i of an outcome o is $v_i(o, \theta_i) = -\sum_{t \in T_i(o)} C_i(t)$. The center's tentative value of an outcome o is $V(o) = \sum_{t \in T_A(o)} R(t)$. Given an outcome o, its social welfare - considering the center and the agents - is $SW(o) = V(o) + \sum_{i \in \alpha} v_i(o, \theta_i)$. Alternatively, the social welfare $SW(o)$ can be viewed as the summation of the social welfare of each assigned task in o, i.e., $SW(o) = \sum_{t \in T_A(o)} SW(t)$, where $SW(t) = R(t) - C_i(t)$ is the social welfare from assigning task t to agent i. Based on the vector $\theta = (\theta_1, \ldots, \theta_n)$ of the agents' reported types, the center will determine an efficient outcome $o_d \equiv o_d(\theta)$ from the set O of all possible outcomes.

Definition 1. *The determined outcome o_d is* efficient *if o_d maximizes the social welfare, i.e., $o_d = argmax_{o \in O} SW(o)$, and $SW(o_d) \geq 0$.*

Under o_d, each task t is simply assigned to agent i who can perform it for the cheapest cost (i.e., highest $SW(t)$), given that the predecessor tasks $t_{\prec}$ of t are assigned.

Utilities and Mechanism Design. Given the determined efficient outcome o_d, the center pays each agent i a payment $p_i(o_d)$ for its contributions in o_d. Assuming quasi-linear utilities, the utility of agent i is $u_i(o_d, \theta_i) = v_i(o_d, \theta_i) + p_i(o_d)$, while the center's utility is $U(o_d, \theta) = V(o_d) - \sum_{i \in \alpha} p_i(o_d)$. To guarantee the efficiency of o_d, the center must propose a payment schema $p_i(o_d)$ for each agent i that guarantees that the agent reports its private type truthfully. Clearly, this is a mechanism design problem [Mas-Colell *et al.*, 1995]. We will focus our attention here on *direct revelation* (DR) mechanisms, where an agent reports *all* its private information to the center that determines o_d and organizes payments to the agents. The revelation principle states that the properties of any mechanism can be replicated by a DR mechanism, and thus, any obtained results here immediately generalize to other indirect mechanisms. The mechanism needs *primarily* to establish truthfulness under some solution concept (Definition 2), either in *dominant strategies* (i.e., *strategy-proof*) or in *ex-post incentive compatibility*. Dominant strategy implementation is the strongest and most preferable solution concept, as it ensures that an agent reports truthfully irrespective of other agents' behavior.

Definition 2. *Given a true type θ_i of agent i, a strategically misreported type $\theta_i^{'}$ of agent i, a vector of reported types $\theta_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n)$ of other agents except agent i, an outcome o_d that is determined if agent i reports θ_i, and an outcome $o_d^{'}$ that is determined if agent i reports $\theta_i^{'}$, a DR mechanism achieves* truthfulness *in*

Dominant Strategy: *For any agent i, reporting truthfully is always an optimal strategy regardless of whether other agents are reporting truthfully or not, i.e., $\forall i \in \alpha$, $u_i(o_d, \theta_i) \geq u_i(o_d^{'}, \theta_i)$ for any reported θ_{-i}.*

Ex-Post Incentive Compatibility: *For any agent i, reporting truthfully is always an optimal strategy given that other agents are reporting truthfully, i.e., $\forall i \in \alpha$, $u_i(o_d, \theta_i) \geq u_i(o_d^{'}, \theta_i)$ given that θ_{-i} holds the true types of other agents.*

DR mechanisms are preferred to possess other properties such as *individual rationality* and *center rationality*.

Definition 3. *A DR mechanism is* individually rational *if for every truthful agent i, its participation guarantees it a non-negative utility (i.e., $u_i(o_d, \theta_i) \geq 0$) given any $o_d \in O$.*

Definition 4. *A DR mechanism is* center rational *if in the truth-telling equilibrium, the center has a non-negative utility (i.e., $U(o_d, \theta) \geq 0$) given any outcome $o_d \in O$.*

Strategic Misreporting. Recalling the type $\theta_i = \langle T_i; C_i(t), \forall t \in T_i \rangle$ of agent i, agent i may increase its utility by strategically misreporting its type to the center in the following three ways: *1.* over-report its ability to perform more tasks than its actual ability (i.e., over-report $T_i^{'} \supset T_i$), this implies a larger set of outcomes $O^{'} \supset O$ from which the center will determine the problem's outcome; *2.* under-report its ability to perform tasks than its can actually perform (i.e., under-report $T_i^{'} \subset T_i$), this implies a smaller set of outcomes $O^{'} \subset O$; and *3.* misreport different costs for performing tasks than the actual costs (i.e., misreport cost $C_i^{'}(t) \neq C_i(t)$ for any task $t \in T$), this implies that the same assignments in $O^{'}$ and O may correspond to different social welfare.

Failures, Executed Outcome and Actual Values. Given the possibility that agents may over-report, we define a *failure point* as a task that wasn't executed successfully. Given any possible failing task, all its successor tasks will *not* be executed. We denote o_e as the part of the determined outcome o_d that was successfully executed, $T(o_e)$ as the set of successful tasks, and $T_i(o_e)$ as the set of successful tasks executed by agent i. Given the possibility that the unexecuted tasks may include tasks that belong to agent i, the *actual* value of agent i is $v_i(o_e, \theta_i) = -\sum_{t \in T_i(o_e)} C_i(t)$, which may differ from its tentative value $v_i(o_d, \theta_i) = -\sum_{t \in T_i(o_d)} C_i(t)$. We define $T_f(o_d)$ as the set of tasks that weren't executed successfully (i.e., $T_f(o_d) = T_A(o_d) - T(o_e)$), which includes all failure points and their successor tasks, and we define $T_f^i(o_d) \subseteq T_f(o_d)$ as the set of tasks that were assigned to agent i and weren't executed because of preceding failures.

Interdependent Valuations and Center rationality. Our problem differs from a classical mechanism design problem in two main aspects. *First, interdependent valuations.* Classical mechanism design normally assumes that the value $v_i(o_d, \theta_i)$ of an agent i of o_d depends *only* on its type θ_i (i.e., independent valuations). But here valuations are interdependent, as the actual value $v_i(o_e, \theta_i)$ of agent i clearly depends on its type, and the *actual* types of other agents who may cause execution failures (i.e., $o_d \neq o_e$). *Second, center rationality.* Classical mechanism design usually assumes that the central authority that determines the problem's outcome is an unbiased party that has no self-interests, as it solves a *social choice* problem that involves *only* the agents. And thus, it is not preferred that this authority ends up with any left-over from the agents' payments (i.e., happens if a weakly

budget balanced mechanism is used), and redistributing the left-over using redistribution mechanisms is required. Here, we assume a commercial ITA model, where the center has its value of the determined outcome, and if center rationality holds, any left-over contributes toward the center's utility (i.e., profit). Thus, we follow Porter et al. [2008] in denoting budget balance as center rationality to point out this issue.

Investment Example. An investment company wants to improve the suitability of a piece of land for construction in order to sell it for a higher price. Possible interdependent tasks for the land improvement are site clearing, removal of trees, general excavation, installation of sewer lines, etc. Assume that the company decided on seven tasks that have the interdependencies $t_1 \prec t_2 \prec \ldots \prec t_7$. The company gets a reward of 10 from each completed task (i.e., the land's price increases by 10 after each task), and wants to assign the tasks to two contactors i and j.

Table 1: Investment Example

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
θ_i	3	6	6	7	4	∞	6
θ_j	∞	4	∞	5	8	∞	3
$\theta_j^{'}$	5	4	5	5	8	∞	3
$\theta_j^{''}$	15	4	5	5	8	7	3

Table 1 includes the contractors' true types θ_i and θ_j, and two misreported types $\theta_j^{'}$ and $\theta_j^{''}$ for contractor j. We use ∞ to denote the inability to perform a task. If $\theta_j^{'}$ was reported, then $o_d = \{(t_1, i), (t_2, j), (t_3, j), (t_4, j), (t_5, i)\}$, where t_3 is assigned to contractor j instead of contractor i if θ_j was reported. Based on o_d, $T_A(o_d) = \{t_1, \ldots, t_5\}$. o_d will fail at task t_3 (because agent j can't execute it), and thus, $T(o_e) = \{t_1, t_2\}$, $T_i(o_e) = \{t_1\}$, $T_f(o_d) = \{t_3, t_4, t_5\}$, and $T_f^i(o_d) = \{t_5\}$.

Non-Negative ITA Model. In this study, we consider a non-negative ITA (NN-ITA) model, where each assigned task must incur a non-negative social welfare (Assumption II). We define our assumptions as follows.

Assumption I. Failure-Detection: *If any task t failed, this failure is detected and the responsible agent is identified.*

This assumption provides a task-by-task monitoring, and is very reasonable when the outcome of a problem is executable. This assumption was used by all similar studies (discussed in Section 5) that deal with outcome failures.

Assumption II. Non-Negative $SW(t)$: *The center will assign a task $t \in T$ only if it incurs a non-negative social welfare, i.e., $SW(t) \geq 0$.*

This assumption narrows down the situations where an efficient outcome is determined, but it is crucial for maintaining the center rationality property. In the general case, the center should assign a task t for a negative social welfare if this will allow assigning its successor tasks, and these successor tasks have a positive social welfare that compensates the negative social welfare of t. This is exactly the same as assuming that the center has combinatorial rewards for the tasks (e.g., gets a single reward of 10 from both t_1 and t_2), and it is proved that achieving center rationality is impossible for combinatorial rewards, even if there are no interdependencies between tasks [Porter *et al.*, 2008, Theorem 4.2].

3 Execution Failures and Sufficient Resources

In this section, we deal only with execution failures assuming that agents have sufficient resources. In other words, given the set of tasks T_i that agent i can perform, the agent has sufficient resources to execute all the tasks assigned for it from T_i. We present this impossibility result.

Theorem 1. *There is no efficient mechanism that achieves strategy-proofness for NN-ITA by using a single allocation round, even if agents have sufficient resources.*

Proof outline. If agent i reported its true type θ_i, then the outcome o_d may: *A.* have a successful execution, or *B.* fail by another agent $j \neq i$. If agent i reported $\theta_i^{'} \neq \theta_i$, then the outcome $o_d^{'}$ may: *1.* have a successful execution, *2.* fail by agent i, or *3.* fail by another agent $j \neq i$. To prove strategy-proofness (Definition 2), we need to show that $u_i(o_d, \theta_i) \geq u_i(o_d^{'}, \theta_i)$ holds in the six possible combinations of A and B respectively with 1, 2 and 3: *Case A1.* Both o_d and $o_d^{'}$ are successful, *Case A2.* o_d is successful and $o_d^{'}$ fails by agent i, *Case A3.* o_d is successful and $o_d^{'}$ fails by another agent $j \neq i$, *Case B1.* o_d fails by another agent $j \neq i$ and $o_d^{'}$ is successful, *Case B2.* o_d fails by another agent $j \neq i$ and $o_d^{'}$ fails by agent i, and *Case B3.* Both o_d and $o_d^{'}$ fail by another agent $j \neq i$. To prove Theorem 1, we prove that there is no payment schema that can cover cases A3 and B1 simultaneously. Let $o_e^{'}$ and $o_f^{'}$ be the executed and unexecuted parts of $o_d^{'}$, respectively.

Proof. Given that the actual value $v_i(o_e, \theta_i) = -\sum_{t \in T_i(o_e)} C_i(t)$ of agent i, the agent's payment $p_i(o_e)$ must increase with each task executed by agent i to compensate the decrease in the agent's value. Any payment schema *either* pays agent i based on only the executed tasks (i.e., $p_i(o_e)$), *or* will also include payments for the unexecuted tasks $T(o_f)$ (i.e., $p_i(o_e, o_f)$). For $p_i(o_e)$, $u_i(o_d, \theta_i) \geq u_i(o_d^{'}, \theta_i)$ will not hold for case B1. This is because agent i may incur some extra costs and prevent the failure [1], which increases the number of executed tasks (i.e., $o_e \subset o_e^{'}$), and thus, its payment. Agent i has incentive to do so if its utility with payment $p_i(o_e^{'})$ will be greater than its utility with payment $p_i(o_e)$. For $p_i(o_e, o_f)$, we want to stress that agent i can by strategic misreporting: 1. make tasks from o_f under o_d belong to $o_e^{'}$ under $o_d^{'}$ (e.g., as in case B1). The agent will do this if the increase in its utility from executed tasks is more than from unexecuted tasks; or 2. make tasks from o_e under o_d belong to $o_f^{'}$ under $o_d^{'}$ (e.g., as [2] in case A3). The agent will do this if the increase in its utility from unexecuted tasks is more than from executed tasks. For $u_i(o_d, \theta_i) \geq u_i(o_d^{'}, \theta_i)$

[1] In the investment example, θ_i and $\theta_j^{'}$ were reported, and o_d will fail by contractor j at t_3. Contractor i can claim t_3 under $o_d^{'}$ by reporting $\theta_i^{'}$ that misreports the cost of t_3 to be 4. Here, $o_d^{'}$ will not fail at t_3, because contractor i can perform t_3, however, for a higher cost than reported.

[2] In the investment example, θ_i and $\theta_j^{'}$ were reported. o_d will fail by contractor j at t_3. Contractor i can report $\theta_i^{'}$ that misreports the cost of t_1 to be 6, and makes t_1 assigned to contractor j under $o_d^{'}$, and $o_d^{'}$ will fail at t_1 because contractor j can't perform it.

to hold for both cases A3 and B1, the increase in the utility of agent i from executed tasks or unexecuted tasks must be the same. To achieve this, $p_i(o_e, o_f)$ must depend directly on the agent's privately known costs for the unexecuted tasks, which can be misreported. □

Reassignment Mechanism. One way to overcome this impossibility result is to design mechanisms that *reassign* failing tasks, i.e., if task t failed, then the center will reassign *only* task t to the agent who reported the second cheapest cost, and then, the execution can start again. The reassignment may happen several time for the same task (e.g., task t failed due to agent i, then reassigned to agent j and failed, then reassigned to agent k and succeeded), and may happen to more than one task. The reassignment will end if all the tasks in o_d were executed successfully, or if there is a *permanent failure* (i.e., a task that failed and can't be reassigned). We define a *temporary failure* as a task that failed and then was executed successfully after reassignment. Using reassignment is very reasonable and common in real-life applications, where the center needs the tasks to be executed. We will now propose a reassignment NN-ITA mechanism, and prove its properties. We denote α_{-i} as the set of agents without agent i, and we denote o_{re} as the executed outcome after the reassignment process. Given the executed outcome o_{re}, we define $SW_{-i}(o_{re})$ as the social welfare of o_{re} without the social welfare of the executed tasks by agent i, i.e., $SW_{-i}(o_{re}) = \sum_{j\in\alpha_{-i}} \sum_{t\in T_j(o_{re})} SW(t)$. As well, we define $SW(o^{-i}(o_{re}))$ as the social welfare of a *virtual* outcome $o^{-i}(o_{re})$, where $o^{-i}(o_{re})$ is the assignment that maximizes the social welfare given the types of other agents $j \neq i$ from the successfully executed tasks in o_{re} (i.e., $T(o_{re})$), while considering Assumption II, neglecting the reported information by agent $j \neq i$ regarding a certain task t if the agent caused its failure, and neglecting the dependencies between the tasks in $T(o_{re})$. *For instance, if θ_i and $\theta_j^{''}$ are reported in the investment example, $o_d = \{(t_1,i),(t_2,j),(t_3,j),(t_4,j),(t_5,i),(t_6,j),(t_7,j)\}$. o_d will fail at t_3, which will be reassigned to contractor i. Then, o_d will fail again at t_6 which is a permanent failure because it can't be reassigned to contractor i. Thus, $o_{re} = \{(t_1,i),(t_2,j),(t_3,i),(t_4,j),(t_5,i)\}$, and $SW_{-i}(o_{re}) = SW(t_2) + SW(t_4) = 6 + 5 = 11$. $SW(o^{-i}(o_{re})) = SW(t_2) + SW(t_4) + SW(t_5) = 6 + 5 + 2 = 13$, because $T(o_{re}) = \{t_1, t_2, t_3, t_4, t_5\}$ and when assigning them to contractor j, t_1 is not assigned because of its negative social welfare, t_3 is not assigned because it failed due to contractor j, and t_2, t_4, t_5 are assigned because we neglected their dependency on t_1 and t_3.*

Definition 5. *A* reassignment NN-ITA mechanism *is defined as follows.*

1. *The center announces the set of the offered tasks T. Then, agents report their types $\theta = (\theta_1, \ldots, \theta_n)$ to the center that will determine an efficient outcome o_d (Definition 1 under Assumption II).*
2. *The outcome o_d then will be executed resulting in o_{re} after reassignments. Each agent i will be paid as follows.*
   a. If agent i caused any temporary or permanent failure, then agent i will get no payment, i.e., $p_i(o_{re}) = 0$.
   b. If the outcome was executed successfully (possibly after reassignment) or permanently failed because of another agent $j \neq i$, then agent i will be paid $p_i(o_{re}) = \sum_{t\in T_i(o_{re})} R(t) + SW_{-i}(o_{re}) - SW(o^{-i}(o_{re}))$.

Theorem 2. *The reassignment NN-ITA mechanism is individually rational for every truthful agent.*

Proof. If agent i caused temporary or permanent failure, then its utility will be

$$u_i(o_d, \theta_i) = -\sum_{t\in T_i(o_{re})} C_i(t), \quad (1)$$

which is negative or 0 if agent i didn't execute any tasks (i.e., $T_i(o_{re}) = \emptyset$). If the execution was successful (possibly after reassignment) or permanently failed due to another agent $j \neq i$, then the utility of agent i will be

$$u_i(o_d, \theta_i) = \sum_{t\in T_i(o_{re})} R(t) - \sum_{t\in T_i(o_{re})} C_i(t) + SW_{-i}(o_{re}) - SW(o^{-i}(o_{re})). \quad (2)$$

For every truthful agent i, its utility is Eq. 2, which can be re-written as $u_i(o_d, \theta_i) = \sum_{t\in T_i(o_{re})} SW(t) + SW_{-i}(o_{re}) - SW(o^{-i}(o_{re})) = SW(o_{re}) - SW(o^{-i}(o_{re}))$. Given that $o_{-i}(o_{re})$ is determined by assigning the executed tasks $T(o_{re})$, then $SW(o_{re}) \geq SW(o^{-i}(o_{re}))$ holds. This is because agent i executes its tasks in o_{re} for the cheapest possible cost (i.e., highest social welfare), but these tasks are assigned in $o^{-i}(o_{re})$ to other agents for higher costs. □

Theorem 3. *The reassignment NN-ITA mechanism is strategy-proof and efficient.*

Proof outline. Considering reassignment, we re-write the six cases in the proof outline of Theorem 1 as follows: *Case A1.* Both o_d and $o_d^{'}$ are successful (possibly after reassignment), *Case A2.* o_d is successful (possibly after reassignment) and any task in $o_d^{'}$ fails temporary or permanently by agent i, *Case A3.* o_d is successful (possibly after reassignment) and $o_d^{'}$ fails permanently by another agent $j \neq i$, *Case B1.* o_d fails permanently by another agent $j \neq i$ and $o_d^{'}$ is successful (possibly after reassignment), *Case B2.* o_d fails permanently by another agent $j \neq i$ and any task in $o_d^{'}$ fails temporary or permanently by agent i, and *Case B3.* Both o_d and $o_d^{'}$ fail permanently by another agent $j \neq i$. To prove strategy-proofness based on Definition 2, we will prove that $u_i(o_d, \theta_i) \geq u_i(o_d^{'}, \theta_i)$ holds in these six cases for any θ_{-i}, given that agent i may practise each type of strategic misreporting (i.e., over-reporting, under-reporting and misreporting costs) separately. By showing that practicing each lying type separately decreases the agent's utility under $o_d^{'}$, then we will have shown any combined strategic misreporting that involves more than one lying type may further decrease the agent's utility under $o_d^{'}$. We *stress* that the payment applies for all the agents who reported their information, and we don't assume that each agent is necessarily assigned tasks under o_d. Once strategy-proofness is established, efficiency follows from step 1 in Definition 5.

Proof. Cases A2 and B2. Under the outcome $o_d^{'}$, the utility of agent i will be negative or 0 (expressed by Eq. 1). However, under the outcome o_d, the agent has

a non-negative utility expressed by Eq. 2 (established in Theorem 2). And thus, $u_i(o_d, \theta_i, \theta_{-i}) \geq u_i(o_d^{'}, \theta_i, \theta_{-i})$ holds. **Cases A1, A3, B1 and B3.** In all the four cases, the utility of agent i under o_d or $o_d^{'}$ is expressed by Eq. 2, and we want to prove that $u_i(o_d, \theta_i) = \sum_{t \in T_i(o_{re})} R(t) - \sum_{t \in T_i(o_{re})} C_i(t) + SW_{-i}(o_{re}) - SW(o^{-i}(o_{re})) \geq u_i(o_d^{'}, \theta_i) = \sum_{t \in T_i(o_{re}^{'})} R(t) - \sum_{t \in T_i(o_{re}^{'})} C_i(t) + SW_{-i}(o_{re}^{'}) - SW(o^{-i}(o_{re}^{'}))$ holds. *Over-reporting:* Given that $o_d^{'}$ is successful (possibly after reassignment) in cases A1 and B1, any over-reported tasks in $\theta_i^{'}$ weren't assigned to agent i. Given that $o_d^{'}$ fails permanently by another agent $j \neq i$ in cases A3 and B3, any over-reported tasks in $\theta_i^{'}$ before the permanent failure point weren't assigned to agent i. Given the previous and that Eq. 2 has no terms related to unexecuted tasks, over-reporting has no effect on the agent's utility. *Under-reporting:* If agent i was the only one capable of performing the task t that it under-reported or report a cost that is higher than the task's reward, then t will not be assigned (no agent can perform it or because of Assumption II) and its successor tasks will not be assigned under $o_d^{'}$. This may decrease the payment that agent i pays the center (i.e., $SW(o^{-i}(o_{re}^{'})) < SW(o^{-i}(o_{re}))$) if the unassigned tasks under $o_d^{'}$ contain tasks that were assigned to other agents $j \in \alpha_{-i}$ under o_d. However, this decrease corresponds to an equal decrease in the agent's received payment from the center (i.e., $SW_{-i}(o_{re}) < SW_{-i}(o_{re}^{'})$). As well, $\sum_{t \in T_i(o_{re})} R(t) - \sum_{t \in T_i(o_{re})} C_i(t) > \sum_{t \in T_i(o_{re}^{'})} R(t) - \sum_{t \in T_i(o_{re}^{'})} C_i(t)$ may hold if the unassigned tasks under $o_d^{'}$ contain tasks that were assigned to agent i under o_d, as any executed task by agent i corresponds to non-negative increase in its utility under Assumption II. *Misreporting costs:* By using reassignment, we **stress** that agent i doesn't need to misreport costs to prevent failures (as in footnote 1), as any failing tasks will be reassigned to agent i or any other agent $j \neq i$ who can execute them successfully. And thus, we can assume that misreporting costs doesn't affect the execution horizon (i.e., $T(o_{re}) = T(o_{re}^{'})$), which implies $SW(o^{-i}(o_{re}^{'})) = SW(o^{-i}(o_{re}))$. Given that $u_i(o_d, \theta_i) = \sum_{t \in T_i(o_{re})} R(t) - \sum_{t \in T_i(o_{re})} C_i(t) + SW_{-i}(o_{re}) = SW(o_{re})$, and $u_i(o_d^{'}, \theta_i) = \sum_{t \in T_i(o_{re}^{'})} R(t) - \sum_{t \in T_i(o_{re}^{'})} C_i(t) + SW_{-i}(o_{re}^{'}) = SW(o_{re}^{'})$, $SW(o_{re}) \geq SW(o_{re}^{'})$ holds because the center initially determines an efficient outcome that maximizes the social welfare, and reassigning failing tasks happens in a manner that maximizes the social welfare (i.e., reassign to the agent who reported the second cheapest cost). □

Theorem 4. *The reassignment NN-ITA mechanism is center rational, and provides profit for the center.*

Proof. In the truth-telling equilibrium, the center pays $p_i(o_{re}) = \sum_{t \in T_i(o_{re})} R(t) + SW_{-i}(o_{re}) - SW(o^{-i}(o_{re}))$ for each agent i. The center's utility of the executed outcome is $U(o_{re}, \theta) = V(o_{re}) - \sum_{i \in \alpha} p_i(o_{re}) = \sum_{t \in T(o_{re})} R(t) - \sum_{i \in \alpha} p_i(o_{re})$, and we need to show that $U(o_{re}, \theta) \geq 0$ holds. The term $\sum_{i \in \alpha} \sum_{t \in T_i(o_{re})} R(t)$ offsets the first term $\sum_{t \in T_i(o_{re})} R(t)$ of each payment $p_i(o_{re})$. Thus, we can represent the center's utility by the remaining terms of each $p_i(o_e)$, i.e., $U(o_{re}, \theta) = \sum_{i \in \alpha} SW(o^{-i}(o_{re})) - SW_{-i}(o_{re})$, and we need to prove that $SW(o^{-i}(o_{re})) \geq SW_{-i}(o_{re})$ holds for each agent i. Recalling that if a task was assigned to agent i, then agent i has the cheapest cost for performing it, and thus, the best social welfare $SW(t)$. Let $SW^{'}(t)$ be the second best social welfare, i.e., assign t to the agent who has the second cheapest cost. $SW(o^{-i}(o_{re})) \geq SW_{-i}(o_{re})$ holds because $SW(o^{-i}(o_{re}))$ contains $SW_{-i}(o_{re})$, in addition to the second best social welfare $SW^{'}(t)$ from each task t that was executed by agent i in o_{re}. This guarantees center rationality, and guarantees that the center gets a lower-bound profit of $SW^{'}(t)$ for each successfully executed task t, given that a second cheapest cost exists. □

4 NN-ITA with Limited Resources

In this section, we assume agents with limited resources, which is adequate for scenarios where acquiring additional resources is not possible. For representing resources, we assume that each agent i has a set of NAND (i.e., negated conjunctions) constraints T_i^{rc} defined over T_i to express the agent's resource constraints (e.g., $t_1, t_2 \in T_i$ and $\neg(t_1 \wedge t_2)$ mean that agent i can't execute both t_1 and t_2 because of limited resources, so the agent may be assigned only t_1, only t_2, or none of them). This representation is suitable because we defined T as a *set* of tasks, which - by definition - doesn't allow the repetition of tasks (e.g., if task t_1 is required to be executed twice, then the second copy must appear under a different notation $t_1^{'}$). Under outcome o_d, the assigned tasks to agent i (i.e., $T_i(o)$) must satisfy the agent's resource constraints (i.e., all constraints in T_i^{rc} must be true). Given that the resource constraints are privately known for agent i, these constraints can be under-reported or over-reported. In limited resources ITA, it is possible to achieve truthfulness in ex-post incentive compatible, but we will not present this result because center rationality is lost and due to space limits as well.

Theorem 5. *There is no mechanism that can achieve center rationality for limited resources NN-ITA, even under ex-post incentive compatible.*

Proof. Assume the following example: *1.* $T = \{t_1, t_2, t_3, t_4\}$ with interdependencies between tasks $t_1 \prec t_2$ and $t_3 \prec t_4$, and each task has a reward of 10; *2.* Two agents i and j; *3.* Agent i is the only agent who can perform t_1 for $C_i(t_1) = 4$ and t_3 for $C_i(t_3) = 2$, but has a resource constraint $\neg(t_1 \wedge t_3)$; *4.* Agent j is the only agent who can perform t_2 for $C_j(t_2) = 1$ and t_4 for $C_j(t_4) = 7$, but has a resource constraint $\neg(t_2 \wedge t_4)$; and *5.* Agent j reports truthfully (i.e., ex-post incentive compatibility). The center can assign either $o_d^1 = \{(t_1, i), (t_2, j)\}$, or $o_d^2 = \{(t_3, i), (t_4, j)\}$. Given that this example assumes no second cheapest cost for tasks (i.e., only one agent who can perform each task), any mechanism that guarantees truthfulness in ex-post incentive compatibility must pay each agent the whole reward of the task it executed. If agent i reported truthfully, then the center will choose o_d^1 (i.e., $SW(o_d^1) = 15 > SW(o_d^2) = 11$), and

the utility of agent i will be $10 - 4 = 6$. Here, agent i can under-report the ability to perform t_1 (i.e., excludes o_d^1). This makes the center chooses the only remaining outcome o_d^2, and the utility of agent i will be $10 - 2 = 8$. To prevent that from happening, the center must pay agent i an amount more than the reward of t_1, and given that the center pays agent j the whole reward for t_2, then center-rationality is lost. □

This impossibility result finalizes our study, as center rationality is a crucial property for mechanisms proposed for commercial use. Maintaining center rationality as well as achieving truthfulness in dominant strategy for limited resources NN-ITA is possible by imposing assumptions (e.g., cost verification as in [Porter *et al.*, 2008]).

5 Discussion and Related Work

Interdependent Valuations. We stress that outcome failure problems (e.g., task allocation, multiagent planning) are not the only type of problem that involves interdependent valuations (see [Mezzetti, 2004] for other examples), and if tasks are not interdependent (i.e., independent valuations), strategy-proof mechanisms already exist (e.g., [Nisan and Ronen, 2001]). When valuations are interdependent, a Groves mechanism [Groves, 1973] loses its strategy-proofness, because its payment depends on the agents' tentative values. All previous efficient mechanisms for interdependent valuations settings achieve truthfulness at ex-post incentive compatibility. Mezzetti [2004] introduced a two-stage Groves mechanism, which works for any interdependent valuations problem. This mechanism is identical to a Groves mechanism, except for a second reporting phase, where agents report their actual values of the determined outcome, and the Groves payment is made based on these actual values. This second reporting phase can be eliminated under Assumption I, as the center is monitoring the outcome and knows the agents' actual values. Domain specific mechanisms for outcome failure problems can handle failures easily, as agent i can be the *only agent* behind the outcome failure (i.e., other agents are reporting truthfully under ex-post incentive compatibility). In [Porter *et al.*, 2008; Ramchurn *et al.*, 2009], mechanisms were proposed for task allocation, where valuations were interdependent in the first because of the interdependencies between tasks, while in the second because of assuming a trust-based model. In [van der Krogt *et al.*, 2008; Zhang and de Weerdt, 2009], mechanisms were proposed for multiagent planning, where valuations were interdependent because of the interdependencies between the plans executed by different agents. The multiagent planning model is more complicated than an ITA model, as interdependencies between actions are not pre-defined, and agents report their own goals and the goals' associated rewards.

Failure Models. Previous studies assume that an outcome may fail either accidentally (e.g., [Porter *et al.*, 2008]) by assuming that an agent privately knows its probability of success (PoS) of $[0, 1]$ when performing a particular task, or intentionally (e.g., [Zhang and de Weerdt, 2009]) as we assume here (i.e., an agent reports PoS of '1' for a task instead of reporting its true PoS of '0'). Accidental failure models assume that a task may fail even if the agent reported truthfully its PoS, and an agent will attempt a task only once. To extend the work proposed here to consider accidental failures, we need to differentiate between if an agent failed because it can't execute the task at all (where the task must be reassigned to another agent as we did here), and between if the agent can execute the task but failed because there is a PoS (where here the agent must keep trying to execute the task until it succeeds). We can achieve this differentiation by extending Assumption I to allow the center to decide whether an agent attempted to execute a task in the first place or not, and we leave that for future work.

Private Durations. Another way - a study we have under review - to design strategy-proof mechanisms for ITA without using reassignment is to factorize the agent's privately known cost for performing a task into two components: a privately known duration in which the agent can perform that task, and a publicly known unit cost associated with each duration unit. Although here and previous studies [Porter *et al.*, 2008; Ramchurn *et al.*, 2009; Zhang and de Weerdt, 2009] use Assumption I, assuming private durations gives an additional advantage, because if an agent claims the ability to perform a task in a shorter period than its actual capability, then the agent can easily be detected. With private costs, an agent can execute a task for a higher or lower cost than its actual cost without being detected.

6 Conclusions and Future Work

In this study, we proposed a reassignment mechanism that is efficient and strategy-proof when valuations are interdependent. And then, we illustrated the effects of assuming agents with limited resources. Interdependent valuations introduce a lot of complexities to the classical mechanism design problem, which only can be handled by designing domain specific mechanisms. Extending the current model and methods to consider combinatorial values in ITA, and to multiagent planning appear fruitful avenues of pursuit.

References

[Groves, 1973] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[Mas-Colell *et al.*, 1995] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford Uni. Press, 1995.

[Mezzetti, 2004] C. Mezzetti. Mechanism design with interdependent valuations: Efficiency. *Econometrica*, 72(5), 2004.

[Nisan and Ronen, 2001] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35, 2001.

[Porter *et al.*, 2008] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Fault tolerant mechanism design. *Artificial Intelligence*, 172(15):1783–1799, 2008.

[Ramchurn *et al.*, 2009] S. D. Ramchurn, C. Mezzetti, A. Giovannucci, J. A. Rodriguez-Aguilar, R. K. Dash, and N. R. Jennings. Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty. *Journal of Artificial Intelligence Research*, 35:119–159, 2009.

[van der Krogt *et al.*, 2008] R. van der Krogt, M. M. de Weerdt, and Y. Zhang. Of mechanism design and multiagent planning. In *The 18th ECAI*, 2008.

[Zhang and de Weerdt, 2009] Y. Zhang and M. M. de Weerdt. Creating incentives to prevent intentional execution failures. In *IEEE/WIC/ACM*, pages 431–434, 2009.

Compact Representation Scheme of Coalitional Games Based on Multi-terminal Zero-suppressed Binary Decision Diagrams

Ryo Ichimura, Yuko Sakurai, Suguru Ueda, Atsushi Iwasaki, Makoto Yokoo
Kyushu University
{ichimura@agent., ysakurai@, ueda@agent., iwasaki@, yokoo@}inf.kyushu-u.ac.jp
Shin-Ichi Minato
Hokkaido University
minato@ist.hokudai.ac.jp

Abstract

Coalitional games, including Coalition Structure Generation (CSG), have been attracting considerable attention from the AI research community. Traditionally, the input of a coalitional game is a black-box function called a characteristic function. Previous studies have found that many problems in coalitional games tend to be computationally intractable in this black-box function representation. Recently, several concise representation schemes for a characteristic function have been proposed. Among them, a synergy coalition group (SCG) has several good characteristics, but its representation size tends to be large compared to other representation schemes.

We propose a new concise representation scheme for a characteristic function based on a Zero-suppressed Binary Decision Diagram (ZDD) and a SCG. We show our scheme (i) is fully expressive, (ii) can be exponentially more concise than the SCG representation, (iii) can solve core-related problems in polynomial time in the number of nodes, and (iv) can solve a CSG problem reasonably well by utilizing a MIP formulation. A Binary Decision Diagram (BDD) has been used as unified infrastructure for representing/manipulating discrete structures in such various domains in AI as data mining and knowledge discovery. Adapting this common infrastructure brings up the opportunity of utilizing abundant BDD resources and cross-fertilization with these fields.

1 Introduction

Forming effective coalitions is a major research challenge in AI and multi-agent systems (MAS). A coalition of agents can sometimes accomplish things that individual agents cannot or can do things more efficiently. There are two major research topics in coalitional games. The first involves partitioning a set of agents into coalitions so that the sum of the rewards of all coalitions is maximized. This is called the Coalition Structure Generation problem (CSG) [Sandholm *et al.*, 1999]. The second topic involves how to divide the value of the coalition among agents. The theory of coalitional games provides a number of solution concepts.

Previous studies have found that many problems in coalitional games, including CSG, tend to be computationally intractable. Traditionally, the input of a coalitional game is a black-box function called a characteristic function that takes a coalition as an input and returns its value. Representing an arbitrary characteristic function explicitly requires $\Theta(2^n)$ numbers, which is prohibitive for large n.

Recently, several concise representation schemes for a characteristic function have been proposed [Conitzer and Sandholm, 2006; Elkind *et al.*, 2008; Ieong and Shoham, 2005; Shrot *et al.*, 2010]. Among them, the synergy coalition group (SCG) [Conitzer and Sandholm, 2006] has several good characteristics. However, a SCG tends to require more space than other representation schemes such as marginal contribution networks [Ieong and Shoham, 2005].

In this paper, we propose a new concise representation scheme for a characteristic function, based on the idea of *Binary Decision Diagram* (BDD) [Akers, 1978]. A BDD is graphical representations that can compactly represent a boolean function. We use ae variant of BDD called a Zero-suppressed BDD (ZDD) [Minato, 1993] that can compactly represent a set of combinations. More specifically, we use a Multi-Terminal ZDD (MTZDD), which can compactly represent a SCG. This representation preserves the good characteristics of a SCG. The following are the features of our scheme: (i) it is fully expressive, (ii) it can be exponentially more concise than a SCG, (iii) such core-related problems as core-non-emptiness, core-membership, and finding a minimal non-blocking payoff vector (cost of stability) can be solved in polynomial time in the number of nodes in a MTZDD, and (iv) although solving a CSG is NP-hard, it can be solved reasonably well by utilizing a MIP formulation.

A BDD was originally developed for VLSI logic circuit design. Recently, A BDD has been applied to various domains in AI, including data mining and knowledge discovery . In these domains, we need to handle logic functions or combination sets efficiently. A BDD has been used as unified infrastructures for representing/manipulating such *discrete structures*. A vast amount of algorithms, software, and tools related to a BDD already exist, e.g., an arithmetic boolean expression manipulator based on a BDD, and a programs for calculating combination sets based on a ZDD [Minato,

1993]. Adapting this common infrastructure for coalitional game theory brings up the opportunity to utilize these abundant resources and for cross-fertilization with other related fields in AI.

2 Preliminaries

2.1 Coalitional Games

Let $A = \{1, 2, \ldots, n\}$ be the set of agents. Since we assume a characteristic function game, the value of coalition S is given by characteristic function v, which assigns a value to each set of agents (coalition) $S \subseteq A$. We assume that each coalition's value is non-negative.

Coalition structure CS is a partition of A into disjoint and exhaustive coalitions. To be more precise, $CS = \{S_1, S_2, \ldots\}$ satisfies the following conditions: $\forall i, j\ (i \neq j),\ S_i \cap S_j = \emptyset, \bigcup_{S_i \in CS} S_i = A$. The value of coalition structure CS, denoted as $V(CS)$, is given by: $V(CS) = \sum_{S_i \in CS} v(S_i)$. Optimal coalition structure CS^* is a coalition structure that satisfies $\forall CS, V(CS^*) \geq V(CS)$.

We say a characteristic function is super-additive, if for any disjoint sets S_i, S_j, $v(S_i \cup S_j) \geq v(S_i) + v(S_j)$ holds. If the characteristic function is super-additive, solving CSG becomes trivial; the grand coalition is optimal. We assume a characteristic function can be non-super-additive.

The core is a prominent solution concept focusing on stability. When a characteristic function is not necessarily super-additive, creating a grand coalition does not make sense. As discussed in [Aumann and Dreze, 1974], we need to consider the stability of a coalition structure. The concept of the core can be extended to the case where agents create an optimal coalition structure. Assume $\pi = (\pi_1, \ldots, \pi_n)$ describes how to divide the obtained reward among agents. We call π a *payoff vector*.

Definition 1 *The core is the set of all payoff vectors π that satisfy the feasibility condition: $\sum_{i \in A} \pi_i = V(CS^*)$, and non-blocking condition: $\forall S \subseteq A, \sum_{i \in S} \pi_i \geq v(S)$.*

If for some set of agents S, the non-blocking condition does not hold, then the agents in S have an incentive to collectively deviate from CS^* and divide $v(S)$ between themselves. As discussed in [Airiau and Sen, 2010], there are two alternative definitions of the feasibility condition: (i) $\sum_{i \in A} \pi_i = V(CS^*)$, and (ii) $\forall S \in CS^*, \sum_{i \in S} \pi_i = v(S)$. If (ii) holds, then (i) holds, but not vice versa. Condition (ii) requires that no monetary transfer (side payment) exists across different coalitions. However, as shown in [Aumann and Dreze, 1974], if a payoff vector satisfies both condition (i) and the non-blocking condition, it also satisfies condition (ii). Thus, we use condition (i) as the feasibility condition.

In general, the core can be empty. The ϵ-core can be obtained by relaxing the non-blocking condition as follows: $\forall S \subseteq A, \sum_{i \in S} \pi_i + \epsilon \geq v(S)$. When ϵ is large enough, the ϵ-core is guaranteed to be non-empty. The smallest non-empty ϵ-core is called the least core.

Alternatively, we can relax the feasibility condition as follows: $\sum_{i \in A} \pi_i = V(CS^*) + \Delta$. This means that an external party is willing to pay amount Δ as a subsidy to stabilize the coalition structure. The minimal amount of Δ is called the *cost of stability* [Bachrach *et al.*, 2009].

2.2 SCG

Conitzer and Sandholm [2006] introduced a concise representation of a characteristic function called a *synergy coalition group (SCG)*. The main idea is to explicitly represent the value of a coalition only when some *positive* synergy exists.

Definition 2 *An SCG consists of a set of pairs of the form: $(S, v(S))$. For any coalition S, the value of the characteristic function is: $v(S) = \max_{p_S}$ $\{\sum_{S_i \in p_S} v(S_i)\}$, where p_S is a partition of S; all S_is are disjoint and $\bigcup_{S_i \in p_S} S_i = S$, and for all the S_i, $(S_i, v(S_i)) \in SCG$. To avoid senseless cases without feasible partitions, we require that $(\{a\}, 0) \in SCG$ whenever $\{a\}$ does not receive a value elsewhere in SCG.*

If the value of coalition S is not given explicitly in SCG, it is calculated from the possible partitions of S. Using this original definition, we can represent only super-additive characteristic functions. To allow for characteristic functions that are not super-additive, we add the following requirement on the partition p_S: $\forall p'_S \subseteq p_S$, where $|p'_S| \geq 2$, $(\bigcup_{S_i \in p'_S} S_i, v(\bigcup_{S_i \in p'_S} S_i))$ is not an element of SCG.

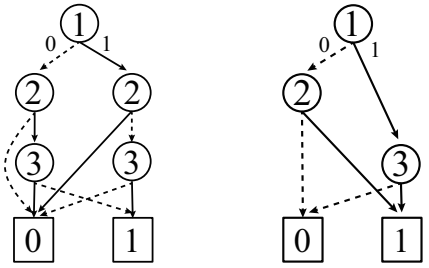
This additional condition requires that if the value of a coalition is explicitly given in SCG, then we cannot further divide it into smaller subcoalitions to calculate values. In this way, we can represent *negative* synergies.

2.3 BDD and ZDD

A BDD represents boolean functions as a rooted, directed acyclic graph of internal nodes and two 0/1-terminal nodes. Each internal node represents a variable and has two outgoing edges: a high-edge and a low-edge. The high-/low-edge means that the value of the variable is true/false. A path from the root node to the 1-terminal node represents that the corresponding value assignment to the path makes the boolean function true. A ZDD is a variant of BDD that can efficiently represent a set of combination. The high-/low-edge means the presence/absence of an element in a combination. In a ZDD, a path from the root node to the 1-terminal node represents that the corresponding value assignment to the path is included in the set.

Consider boolean function $((x_1 \bar{x}_2 x_3) \vee (\bar{x}_1 x_2 \bar{x}_3))$, which can be equivalently represented by using a set of combinations $(\{\{1, 3\}, \{2\}\})$. Figure 1 shows the BDD/ZDD representation for this function/set of combinations. In a tree, a node with x_i represents i. A ZDD is more concise than a BDD. If a variable never appears within any elements in a set of combinations, a node that represents the variable is removed from the ZDD. If the sum of elements contained in all combinations in a set is k, the number of nodes in a ZDD is at most $O(k)$.

Quite recently, two different BDD-based representation schemes for a characteristic function have been developed independently from our work [Aadithya *et al.*, 2011; Berghammer and Bolus, 2010]. While Berghammer and Bolus [2010] deals with simple games, Aadithya *et al.* [2011] considers



(a) BDD (b) ZDD

Figure 1: Examples of a BDD and a ZDD

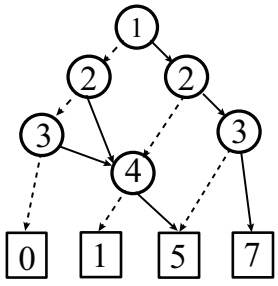


Figure 2: MTZDD representation in Example 1

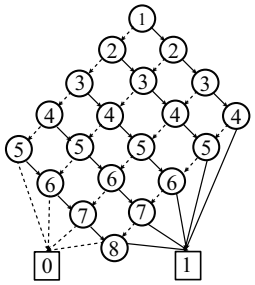


Figure 3: MTZDD representation in Theorem 2

general games. Both of schemes try to represent a characteristic function directly, while our scheme represents SCGs.

3 New Concise Representation Scheme

We propose our new representation scheme for a characteristic function based on a SCG and a ZDD. Although a ZDD can only represent whether a combination exists in a set, a SCG is not just a set of coalitions, because each coalition S in a SCG is associated with its value $v(S)$. Thus, we use a multi-terminal ZDD (MTZDD) representation.

3.1 MTZDD representation based on SCG

A MTZDD G is defined by (V, T, H, L), where V is a set of internal (non-terminal) nodes, T is a set of terminal nodes, H is a set of high-edges, and L is a set of low-edges. Each internal node $u \in V$ is associated with one agent, which we denote as $agent(u)$. u has exactly two outgoing edges, $h(u) = (u, u')$ and $l(u) = (u, u'')$, where $h(u) \in H$ and $l(u) \in L$. Each terminal node $t \in T$ is associated with a non-negative value, which we denote as $r(t)$. Root node u_0 has no incoming edges. For each node $u \in V \setminus \{u_0\} \cup T$, at least one incoming edge exists. We denote the parents of u as $Pa(u)$, $Pa(u) = \{u' \mid (u', u) \in H \cup L\}$.

Path p from root node u_0 to terminal node t is represented by a sequence of edges on path $p = ((u_0, u_1), (u_1, u_2), \ldots, (u_k, t))$. For p, we denote $S(p) = \{agent(u_i) \mid h(u_i) \in p\}$, because $S(p)$ denotes a coalition represented by path p. Also, we denote the value of path p as $r(p)$, which equals $r(t)$: $v(S(p)) = r(t)$. In a MTZDD, a particular ordering among agents is preserved. In path p from root node u_0 to terminal node t, agents associated with nodes in p appear in the same order. More specifically, if node u appears before node u' in p, then $agent(u) \neq agent(u')$. Also, there exists no path p', in which node u appears before node u', where $agent(u) = agent(u')$. For each agent $i \in A$, $nodes(i)$ denotes a set of nodes that are associated with agent i, i.e., $nodes(i) = \{u \mid u \in V \wedge agent(u) = i\}$.

Example 1 *Let there be four agents:* 1, 2, 3, *and* 4. *Let* $SCG = \{ (\{1\}, 1), (\{2\}, 1), (\{3\}, 1), (\{4\}, 0), (\{1, 2\}, 5), (\{1, 4\}, 5), (\{2, 4\}, 5), (\{3, 4\}, 5), (\{1, 2, 3\}, 7)\}$. *This MTZDD representation is described in Figure 2. For example, the rightmost path of the tree represents a coalition* $\{1, 2, 3\}$ *and its value* 7.

3.2 Conciseness of MTZDD Representation

Theorem 1 *MTZDD can represent any characteristic function represented in a SCG using at most* $O(n|SCG|)$ *nodes, where* n *is the number of agents and* $|SCG|$ *is the number of elements in a SCG.*

Proof *In a MTZDD, for each agent* i, $|nodes(i)|$ *is at most* $|SCG|$ *because* $|nodes(i)|$ *represents the number of different contexts that result in different outcomes. This number is bounded by the number of different combinations of agents, which appear before* i *in the ordering among agents. Clearly, this number is at most* $|SCG|$. *Thus, the number of non-terminal nodes, i.e.,* $\sum_{i \in A} |nodes(i)|$, *is at most* $n|SCG|$. *Also, the number of terminal nodes is at most* $|SCG| + 1$. *As a result, the total number of nodes is* $O(n|SCG|)$. □

Theorem 2 *A MTZDD representation is exponentially more concise than a SCG for certain games.*

Proof *Consider a coalitional game with* $2m$ *agents, where the value of characteristic function* $v(S)$ *is* 1 *if* $|S| \geq m$, *and* 0 *otherwise. A SCG must include each coalition with size* m. *The number of such coalitions is given as* $\binom{2m}{m}$, *which is* $O(2^n)$ *using Stirling's approximation.*

On the other hand, we can create a MTZDD that counts the number of agents in a coalition and returns 1 *when the number reaches* m. *Such MTZDD requires* $m(m+1)$ *nodes, i.e.,* $O(n^2)$. □

As shown in the proof of Theorem 2, when some agents are *symmetric*, the MTZDD representation can be much more concise than a SCG. Figure 3 shows a MTZDD when we set $m = 4$. The number of nodes is 20, but a SCG requires 70 coalitions.

Instead of representing a SCG with a MTZDD, we can directly represent a characteristic function using a MTBDD (such an approach is considered in [Aadithya *et al.*, 2011; Berghammer and Bolus, 2010]). In a MTBDD, an agent that does not appear in a path is considered irrelevant; if $v(S \cup \{i\}) = v(S)$, we only need to describe S in a MTBDD[1]. Thus, we can reduce the representation size to a certain extent by using a MTBDD. However, this MTBDD representation for a characteristic function is not as concise as the MTZDD representation. The following theorem holds.

[1]Note that such an irrelevant agent is not included in a SCG.

Theorem 3 *A MTZDD representation of a SCG is always as concise as a MTBDD representation of a characteristic function. Also, it is exponentially more concise than a MTBDD representation for certain games.*

Proof *The worst case occurs when a SCG contains all possible coalitions. In this case, the representation sizes of the MTZDD and MTBDD are the same.*

Then, we show the case where the MTZDD representation is exponentially more concise. Consider a coalitional game with agents $1, 2, \ldots, n$*, where* $v(\{i\}) = 2^i$*, and* $v(S) = \sum_{i \in S} v(\{i\})$*.* $v(S)$ *can take any integer value from* 1 *to* $2^{n+1} - 1$*. Thus, the number of terminal nodes in the MTBDD becomes* $O(2^n)$*. On the other hand, the number of elements in a SCG is* n*, the number of internal nodes in the MTZDD is* n*, and the number of terminal nodes is* $n + 1$*. Thus, the total number of nodes is* $O(n)$*.* □

3.3 Procedure of constructing a MTZDD representation

Let us consider how a person, who has knowledge of a coalitional game, can describe our MTZDD representation. We assume the person is aware of symmetry among agents. Then, the person first describe several partial MTZDDs considering the symmetry among agents. For example, if a person is describing the characteristic function used in the proof of Theorem 2, we can assume she describes multiple partial MTZDDs, each of which corresponds to coalitions of k agents (where k varies from m to $2m$). Note that each partial MTZDD can correspond to multiple (possibly exponentially many) items in a SCG. Then, these partial MTZDDs are integrated into a single MTZDD by applying a Union operation [Minato, 1993] and reduction rules described in Section 2.3.

4 Coalition Structure Generation

We propose a new mixed integer programming formulation for solving a CSG problem in the MTZDD representation. In our MTZDD representation, a path from the root node to a terminal node represents a coalition that is included in a SCG. We define a condition where a set of paths, i.e., a set of coalitions, is *compatible*.

Definition 3 *Two paths,* p *and* p'*, are compatible if* $S(p) \cap S(p') = \emptyset$*. Also, set of paths* P *is compatible if* $\forall p, p' \in P$*, where* $p \neq p'$*,* p*, and* p' *are compatible.*

Finding optimal coalition structure CS^* is equivalent to finding set of paths P^*, which is compatible, and $\sum_{p \in P^*} r(p)$ is maximized. We show that P^* is NP-complete and inapproximable.

Theorem 4 *When the characteristic function is represented as a MTZDD, finding an optimal coalition structure is NP-hard. Moreover, unless P = NP, there exists no polynomial-time* $O(|SCG|^{1-\epsilon})$ *approximation algorithm for any* $\epsilon > 0$*.*

Proof *The maximum independent set problem is to choose* $V' \subseteq V$ *for a graph* $G = (V, E)$ *such that no edge exists between vertices in* V'*, and* $|V'|$ *is maximized under this constraint. It is NP-hard, and unless* $\mathcal{P} = \mathcal{NP}$*, there exists no polynomial-time* $O(|V|^{1-\epsilon})$ *approximation algorithm for any* $\epsilon > 0$ *[Håstad, 1999]. We reduce an arbitrary maximum independent set instance to a CSG problem instance as follows. For each* $e \in E$*, let there be agent* a_e*. For each* $v \in V$*, we create an element of SCG, where the coalition is* $\{a_e \mid e \ni v\}$ *and its value is* 1*. Thus, two coalitions have a common element only if they correspond to neighboring vertices. Coalition structures correspond exactly to independent sets of vertices. Furthermore, we transform this SCG representation to a MTZDD representation in polynomial time [Minato, 1993]. As a result, the number of internal nodes in a MTZDD is at least* $|E|$ *and at most* $2|E|$*, since an agent appears in exactly two coalitions.* □

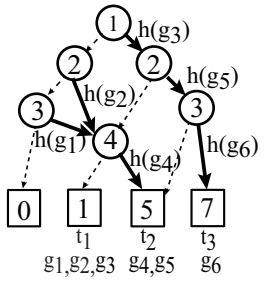


Figure 4: GS in Example 2

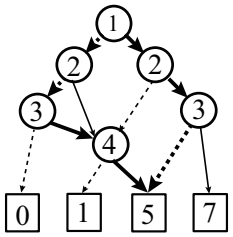


Figure 5: P^* in Example 2

Ohta *et al.* [2009] developed a MIP formulation for a CSG problem when a characteristic function is represented by a SCG. If we enumerate paths, we can use their results. However, the number of paths can be exponential to the number of nodes in a MTZDD. Thus, we need to find P^* without explicitly enumerating all possible paths. We first identify the maximal number of paths within P^*, which leads to one terminal node $r(t)$, using a concept called *minimal required high-edge set* that is concisely described *minimal set*.

Definition 4 *For each terminal node* $t \in T$*, where* $r(t) > 0$*,* $E \subseteq H$ *is a required high-edge set if for all paths* p*, where* t *is* p*'s terminal node, there exists* $h \in E$ *such that* h *is included in* p*.* E *is a minimal set, if* E *is a required high-edge set, and there exists no proper subset of* E *that is a required high-edge set.*

There can be multiple minimal sets. We can find one minimal set using backtrack search starting from the terminal node. The complexity of this procedure is $O(|V|)$. We denote one minimal set of t as $min(t)$. It is clear that the number of paths within P^*, which leads to terminal node $r(t)$, is at most $|min(t)|$.

A MIP formulation of finding P^* is defined as follows. We define some terms and notations. For each terminal node t, where $r(t) > 0$, we create one goal for each element in $min(t)$ and denote the set of goals created from t as $goals(t)$. For each goal $g \in goals(t)$, we denote the corresponding element in $min(t)$ as $h(g)$ and the value of g as $r(g)$, which equals $r(t)$. Let $GS = \bigcup_{t \in T | r(t) > 0} goals(t)$. For each $g \in GS$, $x(g)$ is a 0/1 decision variable that denotes whether g is active ($x(g) = 1$ means g is active). For each goal $g \in GS$ and for each edge (u, u'), $x(g, (u, u'))$ is a 0/1 decision variable that denotes that the edge (u, u') is used for goal g.

Definition 5 *The problem of finding P^* can be modeled as follows.*

$$\begin{aligned}
&\max \textstyle\sum_{g\in GS} x(g)\cdot r(g)\\
&\quad s.t.\quad \forall g\in GS, x(g)=x(g,h(g)), \text{— (i)}\\
&\qquad \forall t\in T, \text{where } r(t)>0, \forall g\in goals(t),\\
&\qquad\quad x(g)=\textstyle\sum_{u\in Pa(t)} x(g,(u,t)), \text{— (ii)}\\
&\qquad \forall u\in V\setminus\{u_0\}, \forall g\in GS,\\
&\qquad\quad x(g,h(u))+x(g,l(u))\\
&\qquad\quad =\textstyle\sum_{u'\in Pa(u)} x(g,(u',u)), \text{— (iii)}\\
&\qquad \forall i\in A, \textstyle\sum_{u\in nodes(i)}\sum_{g\in GS} x(g,h(u))\le 1, \text{— (iv)}\\
&\qquad x(\cdot),x(\cdot,\cdot)\in\{0,1\}.
\end{aligned}$$

Constraint (i) ensures that if goal g is selected, its required high-edge must be selected. Constraint (ii) ensures if one of its goal g is selected for terminal node t, then an edge must exist that is included in a path for g. Constraint (iii) ensures that for each non-terminal, non-root node, correct paths are created (the numbers of inputs and outputs must be the same). Constraint (iv) ensures that one agent can be included in at most one path. In this MIP formulation, the number of constraints is linear to the number of nodes in a MTZDD.

Example 2 *We consider a MIP problem of a MTZDD representation in Example 1.*

First, we create a minimal set for a non-zero-terminal node. As shown in Figure 4, we denote each non-zero terminal node as t_1, t_2, and t_3 from the left. No high-edge directly points to t_1, but using backtracking search, we find three high-edges labeled $h(g_1)$, $h(g_2)$, and $h(g_3)$ as elements of $min(t_1)$. t_2 has both incoming high-edge and low-edge, and so we obtain $min(t_2)=\{h(g_4),h(g_5)\}$. t_3 only has an incoming high-edge, i.e., $min(t_3)=\{h(g_6)\}$. Thus, we obtain $\{g_1,\ldots,g_6\}$ as GS.

Next, we solve a MIP defined by Definition 5 and obtain optimal set of paths P^* that consists of two paths that represent coalitions $\{1,2\}$ and $\{3,4\}$ (Figure 5). The value of P^* is calculated as 10.

5 Core-related Problems

5.1 Core-Non-Emptiness

By assuming that the value of an optimal coalition structure $V(CS^*)$ is given, checking the core-non-emptiness for CS^* can be done in a polynomial time in the number of nodes in a MTZDD. We represent the payoff of an agent as the distance of its high edge. For terminal node t, its shortest distance to the root node represents the minimal total reward of coalition S, where $v(S)=r(t)$. The non-blocking condition requires that, for each terminal node t, its shortest distance to the root node is at least $r(t)$. Let $dis(u)$ represent the shortest distance from root node u_0 to node u.

Definition 6 *The following LP formulation gives an element in the ϵ-core:*

$$\begin{aligned}
&\min\ \epsilon\\
&\quad s.t.\quad dis(u_0)=0,\\
&\qquad \textstyle\sum_{i\in A}\pi_i=V(CS^*),\\
&\qquad \forall u\in V\setminus\{u_0\}\cup T, \forall u'\in Pa(u),\\
&\qquad\quad dis(u)\le dis(u')+\pi_{agent(u')} \text{— if } (u',u)\in H,\\
&\qquad\quad dis(u)\le dis(u') \qquad\qquad \text{— otherwise,}\\
&\qquad \forall t\in T, dis(t)+\epsilon\ge r(t).
\end{aligned}$$

Theorem 5 *By using a MTZDD representation, determining whether the core is non-empty can be done in polynomial time in the number of nodes in a MTZDD, assuming that the value of an optimal coalition structure $V(CS^*)$ is given.*

Proof *To examine whether the core is non-empty, it is sufficient to check whether a solution of the above LP problem is 0 or less. The LP can be solved in polynomial time in the number of its constraints, which is given as $2|V|+|T|$.* □

5.2 Core-Membership

For given payoff vector π, we need to examine whether π is in the core. Assuming the value of an optimal coalition $V(CS^*)$ is given, checking the feasibility condition is easy. For each terminal node $t\in T$, where $r(t)>0$, similar to checking the core-non-emptiness, the non-blocking condition holds if the shortest path $dis(t)$ from the root node to terminal node t is the value of path $r(t)$ or more.

Theorem 6 *By using a MTZDD representation, determining whether a payoff vector π is in the core can be done in $O(|V|)$ time, assuming the value of an optimal coalition structure $V(CS^*)$ is given.*

Proof *A MTZDD is a single-source directed acyclic graph (DAG). Thus, for each terminal node, we can find the distance from the root node using the DAG-shortest paths algorithm, which requires $O(|V|+|H|+|L|)$ time. In a MTZDD, since each internal node has one high-edge and one low-edge, $|V|=|H|=|L|$ holds. It requires $O(|V|)$ time.* □

5.3 The Cost of Stability

Definition 7 *The following LP formulation gives the cost of stability Δ:*

$$\begin{aligned}
&\min\ \Delta,\\
&\quad s.t.\quad dis(u_0)=0,\\
&\qquad \textstyle\sum_{i\in A}\pi_i=V(CS^*)+\Delta,\\
&\qquad \forall u\in V\setminus\{u_0\}\cup T, \forall u'\in Pa(u),\\
&\qquad\quad dis(u)\le dis(u')+\pi_{agent(u')} \text{— if } (u',u)\in H,\\
&\qquad\quad dis(u)\le dis(u') \qquad\qquad \text{— otherwise,}\\
&\qquad \forall t\in T, dis(t)\ge r(t).
\end{aligned}$$

Theorem 7 *By using a MTZDD representation, the cost of stability can be obtained in polynomial time in the number of nodes in a MTZDD, assuming that the value of optimal coalition structure $V(CS^*)$ is given.*

Proof *The cost of stability can be obtained by solving the above LP formulation. The LP can be solved in polynomial time in the number of its constraints, i.e., $2|V|+|T|$.* □

6 Experimental Evaluations

In order to show that our proposed CSG algorithm is reasonably efficient and scalable, we experimentally evaluate its performance, in comparison with the MIP formulation using a SCG representation [Ohta *et al.*, 2009]. The simulations were run on a Xeon E5540 processor with 24-GB RAM. The test machine ran Windows Vista Business x64 Edition SP2. We used CPLEX 12.1, a general-purpose MIP package.

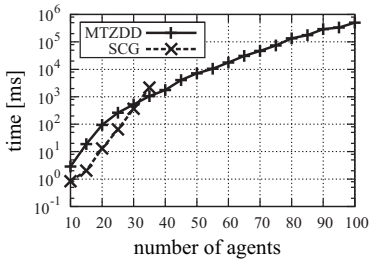


Figure 6: Computation time

We generated problem instances with 5 different groups of symmetric agents. First, we created a set of *abstract rules*. Each rule specifies the required number of agents in each group, which is generated using a decay distribution as follows. Initially, the required number of agents in each group is set to zero. First, we randomly chose one group and incremented the required number of agents in it by one. Then, we repeatedly chose a group randomly and incremented its required number of agents with probability α until a group is not chosen or the required number of agents exceeds the limit ($\alpha = 0.55$) . For each rule, we randomly chose an integer value from $[1, 10]$ as the value of the coalition. The number of abstract rules is set equal to the number of agents. Then, we translated these abstract rules into a MTZDD representation. The MIP formulation using a SCG representation is also generated from these abstract rules. Figure 6 shows the median computation times for solving the generated 50 instances.

When $n \leq 30$, a SCG representation is more efficient than a MTZDD representation for finding an optimal coalition structure, while a MTZDD representation eventually outperforms the SCG for $n > 30$. When the number of coalitions in a SCG is relatively small, the MIP formulation of a SCG representation is simple and CPLEX can reduce the search space efficiently. However, the number of coalitions in a SCG grows exponentially based on the increase of the number of agents/rules. For $n \geq 40$, generating problem instances becomes impossible due to insufficient memory. On the other hand, the number of nodes in a MTZDD grows linearly based on the increase of the number of agents/rules. As a result, the computation time for a MTZDD representation grows more slowly compared to the SCG.

7 Conclusion

We developed a new representation scheme by integrating a ZDD data structure and an existing compact representation scheme called SCG. A ZDD is an efficient data structures applied in various domains in AI. We showed that our MTZDD representation scheme (i) is fully expressive, (ii) can be exponentially more concise than SCG representation, (iii) can solve core-related problems in polynomial time in the number of nodes, and (iv) can solve a CSG problem reasonably well by utilizing a MIP formulation.

Future work includes overcoming the complexity of solving other problems including the Shapley value in coalitional games. We will also consider applying BDD/ZDD-based graphical representation for characteristic functions in non-transferable utility coalitional games.

References

[Aadithya *et al.*, 2011] K. Aadithya, T. Michalak, and N. R. Jennings. Representation of Coalitional Games with Algebraic Decision Diagrams. In *AAMAS*, pages 1121–1122, 2011.

[Airiau and Sen, 2010] S. Airiau and S. Sen. On the stability of an optimal coalition structure. In *ECAI*, pages 203–208, 2010.

[Akers, 1978] S. B. Akers. Binary decision diagrams. *IEEE Transactions on Computers*, C-27(6):509–516, 1978.

[Aumann and Dreze, 1974] R. J. Aumann and J. H. Dreze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3:217–237, 1974.

[Bacchus and Grove, 1995] B. Bacchus and A. Grove. Graphical models for preference and utility. In *UAI*, pages 3–10, 1995.

[Bachrach *et al.*, 2009] Y. Bachrach, R. Meir, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The Cost of Stability and Its Application to Weighted Voting Games. In *SAGT*, pages 122–134, 2009.

[Berghammer and Bolus, 2010] R. Berghammer and S. Bolus. Problem Solving on Simple Games via BDDs. In *COMSOC*, 2010.

[Conitzer and Sandholm, 2006] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6):607–619, 2006.

[Elkind *et al.*, 2008] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. In *AAMAS*, pages 1007–1014, 2008.

[Håstad, 1999] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[Ieong and Shoham, 2005] S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *EC*, pages 193–202, 2005.

[Minato, 1993] S. Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. In *Proc. of the 30th Design Automation Conference (DAC)*, pages 272–277, 1993.

[Ohta *et al.*, 2009] N. Ohta, V. Conitzer, R. Ichimura, Y. Sakurai, A. Iwasaki, and M. Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *CP*, pages 623–638, 2009.

[Sandholm *et al.*, 1999] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238, 1999.

[Shrot *et al.*, 2010] T. Shrot, Y. Aumann, and S. Kraus. On agent types in coalition formation problems. In *AAMAS*, pages 757–764, 2010.

A Liberal Impossibility of Abstract Argumentation

Nan Li

IDEA, Universitat Autònoma de Barcelona, Spain
nan.li@uab.cat

Abstract

In abstract argumentation, where arguments are viewed as abstract entities with a binary defeat relation among them, a set of agents may assign individual members the right to determine the collective defeat relation on some pairs of arguments. I prove that even under a minimal condition of rationality, the assignment of rights to two or more agents is inconsistent with the unanimity principle, whereby unanimously accepted defeat or defend relation among arguments are collectively accepted. This result expands the domain of liberal impossibility beyond preference aggregation and judgment aggregation, and highlights this impossibility as an inherent tension between individual rights and collective consensus.

1 Introduction

Liberal impossibility captures an inherent tension between individual rights and collective consensus. This paper explores whether this impossibility exists in abstract argumentation, a domain different from preference aggregation and judgment aggregation.

In abstract argumentation, a landmark framework introduced by Dung [1995], arguments are viewed as abstract entities with a binary defeat relation among them. Even ignoring the evaluation of the true/false of each argument, there are multiple ways in which an agent may evaluate defeat relations among arguments. Following Sen's [1970] accounts of rights,[1] a set of agents may assign some individual members the right to determine the collective defeat relation on some pairs of arguments.[2] I prove that when only binary evaluation, *i.e.*, true/false, of each argument is permitted, even under a minimal rationality condition, the assignment of rights to two or more agents is inconsistent with the unanimity principle, whereby unanimously accepted defeat or defend relation among arguments, no matter directly or indirectly, are collectively accepted. Thus, liberal impossibility holds.

The discussion on liberal impossibility, or liberal paradox, was ignited by Sen's [1970] seminal paper in the domain of preference aggregation. Outside this domain, Dietrich and List [2008; hereinafter DL] found that this impossibility also exists in the domain of judgment aggregation, and Sen's impossibility can be regarded as a corollary in their framework.

The current work contributes to the classical but in general stagnated debate about individual rights and collective consensus. I prove a liberal impossibility theorem in argumentation, a vast domain but ignored so far by economists, by introducing abstract argumentation into our perspective. I also show that this result is not a corollary of DL's finding, and hence constitutes a complementary work with Sen and DL. In a new domain this result confirms a vague conjecture of Gaertner *et al.* [1992] that "[i]t is our *belief* that this problem[3] persists under virtually every plausible concept of individual rights that we can think of."

The rest of the paper is structured as follows. In Section 2 I provide a very preliminary background of abstract argumentation. I describe the model in Section 3, and prove the impossibility theorem in Section 4. In the last section I briefly present the result of DL [2008], and show that although their work can incorporate Sen's theorem in an extended framework, it fails to do so for the work in the current paper.

2 Abstract Argumentation: Preliminaries

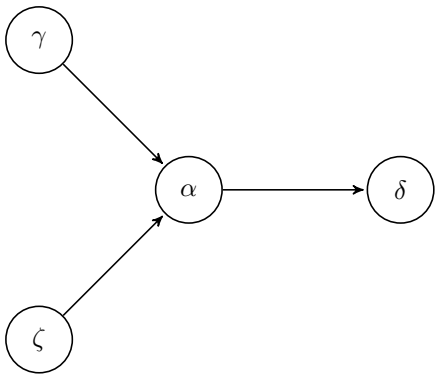
Dung [1995] presented one of the most influential computational models of argumentation. In his model, the internal structure being ignored, arguments are viewed as abstract entities, with a binary defeat relation among them. Formally,

Definition 1 *An* **argumentation framework** *is a pair* $AF = \langle \mathcal{A}, \rightharpoonup \rangle$ *where* $\mathcal{A}$ *is a set of arguments and* $\rightharpoonup$ *is a defeat relation over* $\mathcal{A}$*. We say that an argument* α **defeats** *an argument* β *if* $(\alpha, \beta) \in \rightharpoonup$*, or written as* $\alpha \rightharpoonup \beta$*, and* α *is a* **defeater** *of* β*.*

[1] Sen's paper, especially his formulation of the notion of rights, has encountered different contentions since its publication. For some representative work, see [Nozick, 1974; Gaertner *et al.*, 1992] among others. For recent development, see [Deb *et al.*, 1997; Dowding and van Hees, 2003]. It is not my interest to clarify the notation of rights in the current paper.

[2] For example, some individual members may have expert knowledge on the defeat relation of some pairs of arguments.

[3] That is, the conflict between individual rights and Pareto optimality, a similar concept with unanimity principle here; emphasis and footnote added.



δ: The suspect is innocent according to the presumption of innocence. α: There is evidence that he was at the crime scene *one hour before* the crime. γ: He was witnessed at a nearby town *at the same time* of the crime. ζ: The police obtained evidence that *at that time* he was on the telephone at that town.

Figure 1: *A Murder Case*

For a fixed set of $\mathcal{A}$, in the following section sometimes I use argumentation framework and defeat relation over $\mathcal{A}$ interchangeably to express the same thing if there is no ambiguity.

An argumentation framework can also be represented as a directed graph, *i.e.*, digraph, in which vertices are arguments and the directed arc denotes defeat relation between arguments. An argumentation and its digraph is shown in the following.

Example 2 (A MURDER CASE) *A murder case is under investigation. Initially argument δ states that the suspect is innocent according to the presumption of innocence. But, argument α claims that there is evidence that he was at the crime scene* one hour before *the crime. However, argument γ declares that he was witnessed at a nearby town* at the same time *of the crime. Also, argument ζ asserts that the police obtained evidence that* at that time *he was on the telephone at that town. Argumentation framework $AF = \langle\{\delta, \alpha, \gamma, \zeta\}, \{(\alpha, \delta), (\gamma, \alpha), (\zeta, \alpha)\}\rangle$ corresponds to the digraph in Figure 1.*

In the current work we don't require the defeat relation to be antisymmetric because in real argumentation it is a common phenomenon that two arguments defeat with each other. This is especially usual when we face debates concerning moral value.

Then, when we face an argumentation framework, to determine which arguments are justified and which ones are not is a crucial problem.

For dealing with the reinstatement of arguments, Caminada [2006] introduced the notion of argument labeling, which specifies a particular outcome of argumentation. But for the reason I will mention in the following, here I only adopt his labels `in` and `out`, but not `undec` (undecided).

Definition 3 *Let $\langle \mathcal{A}, \rightharpoonup \rangle$ be an argumentation framework. A* **stable labeling** *is a function $\mathcal{L} : \mathcal{A} \to \{\texttt{in}, \texttt{out}\}$ such that:*

- *$\forall \alpha \in \mathcal{A}$, $\mathcal{L}(\alpha) = \texttt{in}$ if $\mathcal{L}(\beta) = \texttt{out}$ for all β (if any) where $\beta \rightharpoonup \alpha$; and*
- *$\forall \alpha \in \mathcal{A}$, $\mathcal{L}(\alpha) = \texttt{out}$ if there is a β such that $\beta \rightharpoonup \alpha$ and $\mathcal{L}(\beta) = \texttt{in}$.*

With this language, the label `in` means the argument is accepted/justified, the label `out` means the argument is rejected/not justified.

This definition works well for simple cases where we can see clearly which arguments should emerge victoriously. For example, in the argumentation framework $\alpha \rightharpoonup \beta \rightharpoonup \gamma$, α is `in` since it is not defeated by any argument. Consequently β is `out`, and γ is `in`. Even so, however, in some cases the definition above is ambiguous. The Liar Paradox is a famous example that concerns the problem of self-defeat, which makes any determination on which arguments are `in` or `out` impossible based on Definition 3. Thus, if we accept Definition 3, then we impose a constraint on the original definition of argumentation framework, *i.e.*, there is no self-defeating argument. Put in another way, defeat relation $\rightharpoonup$ is irreflexive.

Notice that Definition 3 can actually be seen as a postulate, as it specifies a restriction on both a labeling and an argumentation framework. The meaning of the latter statement will be clear in the following sections.

Definition 4 *Let $\langle \mathcal{A}, \rightharpoonup \rangle$ be an argumentation framework, and $\mathcal{L}$ a labeling over it. We define:*

- $\texttt{in}(\mathcal{L}) = \{\alpha \in \mathcal{A} | \mathcal{L}(\alpha) = \texttt{in}\}$;
- $\texttt{out}(\mathcal{L}) = \{\alpha \in \mathcal{A} | \mathcal{L}(\alpha) = \texttt{out}\}$.

Here an explanation is in order. In the literature of artificial intelligence, starting from the paper of Caminada [2006], many scholars, besides the notion of `in` and `out`, also adopt `undec` to denote the labeling of an argument whose status, *i.e.*, justified or not justified, could not be decided. In real life, *e.g.*, judicial practice, however, an undecided argument is not acceptable. Just as we only call an argument justified or not justified, the labeling of `undec` also is not adopted in the current work. In Section 4 we will see that this refusal is crucial in our impossibility theorem.

We notice that although some argumentation frameworks can only accommodate one stable labeling, say, *e.g.*, argumentation framework $\alpha \rightharpoonup \beta \rightharpoonup \gamma$ with the only stable labeling $\mathcal{L}$ such that $\texttt{in}(\mathcal{L}) = \{\alpha, \gamma\}$, there are many argumentation frameworks which accommodate multiple binary labelings. In fact, suppose there are four arguments where $\alpha \rightharpoonup \beta \rightharpoonup \gamma \rightharpoonup \delta$, and $\delta \rightharpoonup \alpha$, then we see that there exist two binary labelings $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $\texttt{in}(\mathcal{L}_1) = \{\alpha, \gamma\}$ and $\texttt{in}(\mathcal{L}_2) = \{\beta, \delta\}$.

But there exist argumentation frameworks which cannot accommodate at least one stable labeling.

Example 5 *Suppose $\mathcal{A} = \{\alpha, \beta, \delta\}$, and the argumentation digraph is shown as Figure 2. Then, we find that it cannot determine which argument is `in` or `out`. In fact,* e.g.*, if we deem that argument α `in`, then according to Definition 3, argument β is `out`, and argument δ `in`. Consequently, α should not be `out`, a contradiction. The same problem arises when we initially deem that argument α `out`.*

Definition 6 *We call an argumentation framework* **admissible** *if it can accommodate at least one stable labeling; otherwise it is* **inadmissible**.

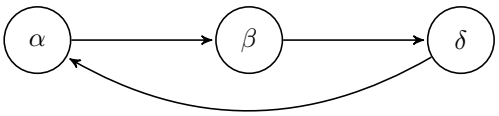


Figure 2: *An Argumentation Framework with Odd Cycle*

For a fixed set of $\mathcal{A}$, we also call the defeat relation over $\mathcal{A}$ admissible or inadmissible depending on the underlying nature of the framework. Just as Definition 3 implies, it does specify a restriction on an argumentation framework. In this paper, I only consider admissible argumentation framework, which captures a minimal condition of rationality, as a reasonable point of view of an agent.

At last, for the definitions that follow, we need to introduce two notations. For any $S \subseteq \mathcal{A}$ and $\alpha \in \mathcal{A}$, let $S^+ = \{\gamma \in \mathcal{A} | \beta \rightharpoonup \gamma \text{ for some } \beta \in S\}$, and $\alpha^- = \{\beta \in \mathcal{A} | \beta \rightharpoonup \alpha\}$.

Definition 7 *Let $\langle \mathcal{A}, \rightharpoonup \rangle$ be an argumentation framework, and let $S \subseteq \mathcal{A}$ and $\alpha \in \mathcal{A}$. We call S* **defends** *argument α if $\alpha^- \subseteq S^+$. We also say that argument α is* **acceptable with respect to** *S.*

Intuitively, a set of arguments defends a given argument if it defeats all its defeaters.[4]

3 The Model: Aggregating Argumentation Framework

In the above section I provide a very preliminary introduction to the element of abstract argumentation, focusing on stable labeling instead of argument labeling with `undec`, or more general one.[5] The choice of contents depends on whether they are relevant to the current research, where we analyze the problem of aggregating different individual argumentation frameworks over a common set of arguments to get a social argumentation framework, and to discuss the inconsistency among some desirable properties.[6] In this section, I introduce the model and define three properties.

Conventionally, we use $\mathbb{N}$ to denote the set of natural numbers. For an integer k, $[k]$ denotes the set $\{1, 2, \ldots, k\}$.

We consider a group of agents $N = [n]$ $(n \geq 2)$, and a finite set of arguments $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ $(m \geq 3)$. For each agent $i \in N$, she has her own argumentation framework $AF_i = \langle \mathcal{A}, \rightharpoonup_i \rangle$, build up from her defeat relation $\rightharpoonup_i$. Given a pair of arguments $\alpha, \beta \in \mathcal{A}$, each agent can express her defeat relation by choosing one of the four alternatives: 1) both arguments are perfectly compatible; 2) α defeats β; 3) β defeats α; or 4) they defeat each other (expressing that they are in conflict but have the same power of argumentation, or are indifferent). If we let $[\alpha, \beta]$ denote any ordered pair of arguments α and β, *i.e.*, $[\alpha, \beta]$ is either (α, β) or (β, α), then in the language of digraph, the four alternatives are: 1) $[\alpha, \beta] \notin \rightharpoonup$; 2) $\alpha \rightharpoonup \beta$; 3) $\beta \rightharpoonup \alpha$; or 4) $\alpha \rightleftharpoons \beta$, respectively.

Bodanza and Auday [2009] provided the following two definitions.[7]

Definition 8 *A* **social defeat function** *is a mapping $f : \rightharpoonup_1 \times \ldots \times \rightharpoonup_n \to \mathcal{A} \times \mathcal{A}$. We call the relation produced by f for each profile of individual defeat relations* **social defeat relation**.

Definition 9 *A* **social argumentation framework** *is a structure $SAF = \langle \mathcal{A}, \{AF_i\}_{i \in N}, \rightharpoonup_f \rangle$, where $\rightharpoonup_f$ is the social defeat relation of SAF produced from social defeat function f.*

Example 10 (SOCIAL ARGUMENTATION FRAMEWORK WITH MAJORITY RULE) *Suppose there are three agents facing a set of three arguments α, β, and γ. Their individual defeat relations are*

$$\rightharpoonup_1 : \ \alpha \rightharpoonup \beta \rightharpoonup \gamma,$$

$$\rightharpoonup_2 : \ \alpha \leftharpoonup \beta \rightharpoonup \gamma,$$

$$\rightharpoonup_3 : \ \alpha \rightharpoonup \beta \leftharpoonup \gamma,$$

respectively. If this society adopts majority rule m as their social defeat function, then the social defeat relation is $\rightharpoonup_m$: $\alpha \rightharpoonup \beta \rightharpoonup \gamma$ $(= \rightharpoonup_1)$.

We can describe the behaviors of defending and defeating with more nuances.

Definition 11 *For any $\alpha, \beta \in \mathcal{A}$, we call α* **indirectly defeats** *β if there exists an $(\alpha, \beta)-$path with length[8] $k = 2l+1$, where $l \in \mathbb{N}$. If not specified explicitly, we write $\alpha \rightsquigarrow \beta$*

[4] Trivially, for any argument α which has no defeater, since $\alpha^- = \emptyset \subseteq \{\alpha\}^+$, $\{\alpha\}$ defends α. In this case, for simplicity we also say that α defends itself.

[5] For an overall summary of the state-of-the-art achievement of this theory, see Rahwan and Simari [2009]. For the relationship between labeling-based approach and extension-based approach when defining argumentation semantics, see Baroni *et al*. [2004].

[6] Different from my concern in the current work, Bodanza and Auday [2009] analyzed the problem of aggregating individual argumentation frameworks over a common set of arguments in order to *obtain a unique socially justified set of arguments* (emphasis added). They articulated the difference of aggregation methods involved. That is, their work "can be done in two different ways: a social attack relation is built up from the individual ones, and then is used to produce a set of justified arguments, or this set is directly obtained from the sets of individually justified arguments." What we do in this research starts from the first step of the first way, although with totally different destination. In contrast, their "main concern here is whether these two procedures can coincide or under what conditions this could happen."

[7] Bodanza and Auday [2009] call *social attack relation* instead of *social defeat relation*, and do not define explicitly social defeat function. Instead, they call the aggregation of individual argumentation frameworks "according to some specified mechanism M."

[8] In digraph D, a *path* is an alternating sequences $P = x_1 a_1 x_2 a_2 x_3 \ldots x_{k-1} a_{k-1} x_k$ of vertices x_i and arcs a_j from D such that the tail of a_i is x_i and the head of a_i is x_{i+1} for every $i \in [k-1]$, and $x_i \neq x_j$ if $i \neq j$, $\forall i, j \in [k]$. We say that P is a path from x_1 to x_k or an (x_1, x_k)-path. The length of a path is the number of its arcs. Hence, the path above has length $k-1$. For P, if $x_1, x_2, \ldots, x_{k-1}$ are distinct, $k \geq 3$ and $x_1 = x_k$, P is a cycle. The length of a cycle is defined in the same way.

no matter α defeats or indirectly defeats β. We call α **indirectly defends**[9] *β if there exists an $(\alpha, \beta)-$path with length $k = 2l + 2$, where $l \in \mathbb{N}$. We write $\alpha \twoheadrightarrow \beta$ no matter α defends or indirectly defends β.*

At the same time, we still need to know that there exits another delicate situation defined below, although it will not be incorporated in the desirable properties for a social defeat function.

Definition 12 *For any $\alpha \in \mathcal{A}$, if there exists $\beta \in \mathcal{A}$ such that $\alpha^- \cap \{\beta\}^+ \neq \emptyset$ but not $\alpha^- \subseteq \{\beta\}^+$, we say that β* **partially defends**[10] *α.*

Intuitively, β partially defends α if α has multiple defeaters, and β defeats some (but not all) of them.

Now, suppose we want to find a social defeat function f with the following intuitive properties:

Universal Domain (Condition D): The domain of f is the set of all profiles where each individual defeat relation is admissible, and the range of f is the set of all defeat relations that is admissible.

Unanimity Principle (Condition U): For any $\alpha, \beta \in \mathcal{A}$, $\alpha \hookrightarrow_f \beta$ if $\alpha \hookrightarrow_i \beta$ for all $i \in N$, and $\alpha \twoheadrightarrow_f \beta$ if $\alpha \twoheadrightarrow_i \beta$ for all $i \in N$.[11]

Minimal Liberalism[12] (Condition L): There are at least *two* agents such that for each of them there is at least one pair of arguments between which she is decisive over the defeat relation. That is, for her there is at least one pair of arguments, say α and β, such that the social defeat relation between these two arguments is the same with her defeat relation between them, *i.e.*, $[\alpha, \beta] \notin \rightharpoonup$, $\alpha \rightharpoonup \beta$, $\beta \rightharpoonup \alpha$, or $\alpha \rightleftharpoons \beta$.

[9] Dung [1995] actually has defined "indirectly defeat" and "indirectly defend". But, using the language here, his called α *indirectly defends* β if there exists an $(\alpha, \beta)-$path with length $k = 2l$, where $l \in \mathbb{N}$. Obviously this definition is not compatible with our definition of "defend" in Definition 7. For example, if there exists an argumentation framework $\beta \rightharpoonup \gamma \rightharpoonup \alpha$, then we see that β *defends* α in our language, but β *indirectly defends* α in Dung's language. Also, in Example 2, if there is another argument β that defeats γ, Dung's definition cannot distinguish the defeat relations among β, γ and α, and $\beta \rightharpoonup \gamma \rightharpoonup \alpha$. Dung would say that in both case β *indirectly defends* α, but I will call β *defends* α in the latter case, and β *partially defends* α in the former case.

[10] Obviously, here $|\alpha^-| > 1$.

[11] We don't impose any constraint on the social defeat relation between any arguments α and β when all agents deem that they are compatible, *i.e.*, $[\alpha, \beta] \notin \rightharpoonup$. Also, there is no constraint when all agents deem argument α partially defends argument β.

Indirect defeat (or defense) can be obtained through different paths for all the agents. Although the paths, that can be seen as different justifications for the statement, are different, we can still think all individuals share a similar opinion when $\alpha \hookrightarrow_i \beta$ for all $i \in N$. That is, they agree that α defeats β directly or indirectly. We can interpret the case of $\alpha \twoheadrightarrow_i \beta$ similarly. I use the term "unanimity" in this sense.

[12] This concept can also accommodate the idea of expert right just as in DL [2008], where some group members may have expert knowledge on certain issues and may therefore be granted the right to be decisive on them. To follow the convention, however, I still use the term here.

4 Impossibility Theorem

The following example provides a good motivation for the current work.

Example 13 (A DEBATE ABOUT MIGRATION OF THE DIRTY INDUSTRIES TO THE LDCS)[13] *Imagine there is a debate in a committee of the World Bank about whether it should encourage more migration of the dirty industries to the LDCs (less developed countries). This committee is constituted of economists Alan and Brenda, who have different opinions about the defeat relations among the following three arguments:*

β: The measurement of the costs of health-impairing pollution depends on the foregone earnings from increased morbidity and mortality. From this point of view a given amount of health-impairing pollution should be done in the country with the lowest cost, which will be the country with the lowest wages. Rational agents in LDCs would accept migration of the dirty industries from developed countries for compensation between the least that agents in LDCs will accept and the most that agents in rich countries will offer. This voluntary agreement is an welfare improvement on both parties.

α: In reality normally LDCs accepts migration of the dirty industries due to their ignorance of the potential danger of pollution.

δ: In reality normally LDCs accepting migration of the dirty industries know the potential danger of pollution. But this voluntary agreement is unfair.

Initially both Alan and Brenda are welfarists who believe that morality is centrally concerned with the welfare or well-being of individuals. Thus, argument β is a counterargument of argument δ. Besides that, Brenda considers that argument α is a counterargument of β, so her argumentation framework is

$$\textit{Brenda: } \alpha \rightharpoonup \beta \rightharpoonup \delta.$$

That is, she is not a stubborn welfarist, and realizes that there are hidden stories behind the so-called "voluntary" agreement. Consequently she prefers to give up her support to argument β, and finally justifies arguments α and δ.

On the contrary, Alan considers that argument δ is a counterargument of α. For him, no matter how to evaluate a policy, in reality there are many agreements where one party has to or prefer to sign even all negative influences involved are known; acceptance of dirty industry is one of these cases. So his argumentation framework is:

$$\textit{Alan: } \beta \rightharpoonup \delta \rightharpoonup \alpha.$$

That is, since he is a stubborn welfarist, unshakably he justifies argument β, and argument α too with the sacrifice of argument δ.[14]

[13] This example is inspired by a shocking real one, see pp.12-23 of Hausman and McPherson [2006].

[14] This is a special case where, according to Definition 3, once we know the argumentation framework of any member of committee, we know her of his evaluation of justified or not for each argument.

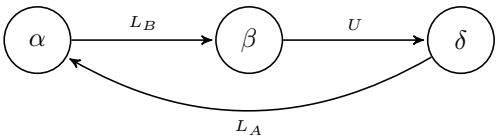


Figure 3: *A Debate about Migration of the Dirty Industries to the LDCs*

Now suppose that Brenda is an expert in dealing with the defeat relation between arguments α and β, so the World Bank assigns her the task. Similarly, Alan is assigned to determine the defeat relation between arguments δ and α. Also, this committee accepts the unanimous defeat relation among any pair of arguments. Under the circumstances, we see that the committee as a whole, its argumentation framework can be depicted as the one in Figure 2. For convenience, we reproduce it in Figure 3.[15]

Then, the committee finds that it cannot determine which argument is justified or not since this is an inadmissible argumentation framework.

The following theorem reveals an inherent tension between liberal rights and collective consensus in a most general situation of argumentation, where the core concepts are only defeating, defending, and (not) being justified.

Theorem 14 *There is no social defeat function that can simultaneously satisfy Conditions D, U, and L in abstract argumentation.*

Proof. Remember that for any pair of arguments, say α and β, an agent can express one of the four alternatives, 1) $[\alpha, \beta] \notin \rightharpoonup$; 2) $\alpha \rightharpoonup \beta$; 3) $\beta \rightharpoonup \alpha$; or 4) $\alpha \rightleftharpoons \beta$, respectively. For any society which respects liberal right, such an alternative should form the social defeat relation between α and β if this agent is decisive over the defeat relation between these two arguments.

Let the two agents referred to in Condition L be 1 and 2, respectively, and the two pairs of arguments referred to be (α, β) and (δ, γ), respectively. There are no more other arguments in $\mathcal{A}$. If (α, β) and (γ, δ) are the same pair of arguments, then there is a contradiction. Thus, they have at most one argument in common, say $\alpha = \gamma$. Assume now that agent 1 deems that α defeats β, and agent 2 deems that δ defeats γ $(= \alpha)$. And let everyone in the community including agent 1 and 2 deem that β defeats δ. That is, the argumentation frameworks of agent 1 and 2 are

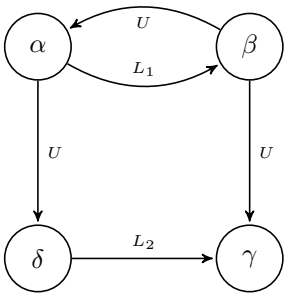
$$\text{agent 1: } \alpha \rightharpoonup \beta \rightharpoonup \delta;$$
$$\text{agent 2: } \beta \rightharpoonup \delta \rightharpoonup \alpha.$$

By Condition D all the frameworks are admissible. But by Condition L, as the society, α must defeat β, and δ must defeat γ $(= \alpha)$, while by Condition U, β must defeat δ. Consequently, we get the same argumentation framework with odd cycle as shown in Figure 3. An argumentation framework with odd cycle, however, obviously is inadmissible, a contradiction.

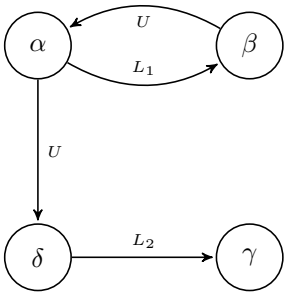
Next, let α, β, γ and δ be all distinct. Besides deeming that α defeats β from her liberal right, suppose that agent 1 also deems that β defeats γ, and γ defeats δ. Let everyone else in the community including agent 2 deem that β defeats α, α defeats δ, and δ defeats γ. That is,

$$\text{agent 1: } \alpha \rightharpoonup \beta \rightharpoonup \gamma \rightharpoonup \delta;$$
$$\text{agent } 2, \ldots, n\text{: } \beta \rightharpoonup \alpha \rightharpoonup \delta \rightharpoonup \gamma.$$

By Condition D all the frameworks are admissible. But by Condition L, as the society, α must defeat β, and remembering the liberal right of agent 2, δ should defeat γ, while by Condition U, α must defeat or indirectly defeat δ. Since we have known that δ defeats γ, it follows that α cannot indirectly defeat δ. Thus, α must defeat δ. Similarly we see that β defeats or indirectly defeats γ. Also, α must defend γ by Condition U, so there is no arc from α to γ. Nevertheless, by Condition U again, β must defend or indirectly defend δ, and since there is only one argument α that defeats δ, consequently β must defeat α. Depending on whether β both defeats and indirectly defeats γ, or β only indirectly defeats γ, the social argumentation frameworks can be shown in Figure 4.[16] No matter in which case, it contradicts with the liberal right of agent 1, who deems that α defeats β. ■



(a) β both defeats and indirectly defeats γ



(b) β only indirectly defeats γ

Figure 4: *A Liberal Paradox of Four Arguments*

From the explanation under Definition 4, however, we should know that this is not universal. In any case, what we are interested in is the aggregation of argumentation frameworks, not the aggregation of individual viewpoint as to the justification of arguments.

[15] In the following digraphs we sometimes label the force determining the social defeat relation between two arguments besides the corresponding arrow, where U denotes Condition U, and L with a subscript denotes the liberty of corresponding economist (agent).

[16] Which argumentation framework is the final one depends on more details about the social defeat function. But this is not the interest of the current research.

5 Discussion: beyond Judgment Aggregation

Liberal impossibility not only haunts preference aggregation, it also appears in judgment aggregation, an emerging active multidisciplinary field. In a recent paper, Dietrich and List (2008) identified a problem that generalizes Sen's liberal paradox. Under plausible conditions, they proved that the assignment of rights to two or more agents or subgroups is also inconsistent with the unanimity principle.

Simply speaking, there is a group of agents $N = [n]$ $(n \geq 2)$ and an agenda, *i.e.*, a non-empty subset X of logic $\mathbf{L}$ expressed as $X = \{p, \neg p : p \in X_+\}$ for a set $X_+ \subseteq \mathbf{L}$ of unnegated propositions on which binary judgments, *i.e.*, yes or no, are made. They call propositions $p, q \in X$ *conditionally dependent* if there exist $p^* \in \{p, \neg p\}$ and $q^* \in \{q, \neg q\}$ such that $\{p^*, q^*\} \cup Y$ is inconsistent for some $Y \subseteq X$ consistent with each of p^* and q^*. The agenda X is *connected* if any two propositions $p, q \in X$ are conditionally dependent. Their main finding is that if and only if the agenda is connected, there exists no aggregation function F generating consistent collective judgment sets that satisfies universal domain, minimal rights and the unanimity principle.[17]

Moreover, after an easy transformation from the question of whether alternative a is strict better than alternative b to the question of whether proposition "alternative a is strict better than alternative b" is true, they proved that the preference agenda is connected. Consequently, Sen's Liberal Paradox becomes a corollary naturally.

Since judgment aggregation and argumentation share some common interests, and both depend on the toolset of logic in a different sense, especially due to the implied seemingly relationship between "connected" agenda and digraph, it may be conjectured that the result of DL will cover our finding in the current paper. But we can show that it is totally not the case.

In fact, although the easy transformation mentioned above helps DL successfully incorporate the domain of preference aggregation into the one of judgment aggregation, a similar practice fails to do so for the sake of abstract argumentation. In my model, for each pair of arguments what really is aggregated is the defeat relations between them among all agents, instead of `in` or `out` of these two arguments. Thus, for any two arguments α and β we first need to ask if we introduce proposition p to denote $\alpha \rightharpoonup \beta$, then what? In DL's paper, actually in the mainstream research of judgment aggregation until now, any proposition only adopts classical two-value logic, viz yes or no. When we talk about the aggregation of defeat relation, for any pair of arguments α and β, there exist four possibilities, viz $[\alpha, \beta] \notin \rightharpoonup$, $\alpha \rightharpoonup \beta$, $\beta \rightharpoonup \alpha$, or $\alpha \rightleftharpoons \beta$. Thus, if we use the language of logic, p should be a proposition in a four-value logic, for which DL's framework cannot cover.

Dietrich [2007] does tackled Arrowian impossibility in a generalized model. But it is still an open question whether liberal paradox exists in general logic.

Therefore, what we do in the current paper is a complementary work with Sen and DL.

[17] Concretely, they define these three properties as:

Universal Domain: The domain of F is the set of all possible profiles of consistent and complete individual judgment sets.

Minimal Rights: There exist (at least) two agents who are each decisive on (at least) one proposition-negation pair $\{p, \neg p\} \subseteq X$.

Unanimity Principle: For any profile $(A_1, \ldots, A_n)$ in the domain of F and any proposition $p \in X$, if $p \in A_i$ for all agents i, then $p \in F(A_1, \ldots, A_n)$, where A_i is the judgment set of agent i.

Acknowledgments

The author gratefully acknowledges support from the Ministerio de Ciencia e Innovación de España through Project ECO2008-04756. The author thanks three anonymous reviewers for insightful comments, and participants of seminars in Madrid, London, Stockholm and Luxembourg for helpful suggestions.

References

[1] Baroni, P., Caminada, M. and Giacomin, M. (2004), An Introduction to Argumentation Semantics. *The Knowledge Engineering Review*, Vol. 00:0, 1-24. DOI: 10.1017/S000000000000000.

[2] Bodanza, G. A. and Auday, M. R. (2009), Social Argument Justification: Some Mechanisms and Conditions for Their Coincidence. In C. Sossai and G. Chemello (eds.), *Lecture Notes in Computer Science: Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Vol. 5590/2009, 95-106, Springer-Verlag, Berlin, Heidelberg.

[3] Caminada, M. (2006), On the Issue of Reinstatement in Argumentation. In M. Fisher,W. van der Hoek, B. Konev, and A. Lisitsa (eds.), *Lecture Notes in Computer Science: Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15, 2006 Proceedings*, Vol. 4160/2006, 111-123, Springer-Verlag, Berlin, Heidelberg.

[4] Deb, R., Pattanaik, P. K. and Razzolini, L. (1997), Game Forms, Rights, and the Efficiency of Social Outcomes. *Journal of Economic Theory*, 72:74-95.

[5] Dietrich, F. (2007), A generalised model of judgment aggregation. *Social Choice and Welfare*, 28:529-565.

[6] Dietrich, F. and List, C. (2008), A Liberal Paradox for Judgment Aggregation. *Social Choice and Welfare*, 31:59-78.

[7] Dowding, K. and van Hees, M. (2003), The Construction of Rights. *American Political Science Review*, 97:281-293.

[8] Dung, P. M. (1995), On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-person Games. *Artificial Intelligence*, 77(2): 321-358.

[9] Gaertner, W., Pattanaik, P. and Suzumura, K. (1992), Individual Rights Revisited. *Economica*, 59:161-77.

[10] Hausman, Daniel M. and McPherson, Michael S. (2006), *Economic Analysis, Moral Philosophy, and Public Policy*, Cambridge University Press.

[11] Nozick, R. (1974), *Anarchy, State and Utopia*, Oxford: Basil Blackwell.

[12] Rahwan, I. and Simari, G. R. (eds.) (2009), *Argumentation in Artificial Intelligence*, Springer, Dordrecht *etc*.

[13] Sen, A. K. (1970), The Impossibility of a Paretian Liberal. *Journal of Political Economy*, 78(1):152-157.

Fair division of indivisible goods under risk

Charles Lumet **Sylvain Bouveret** **Michel Lemaître**
Onera DTIM. 2, avenue Édouard Belin
31055 Toulouse Cedex 4 – FRANCE
first.last@onera.fr

Abstract

We study the problem of fairly allocating a set of indivisible goods to a set of agents having additive preferences. More precisely, we consider the problem in which each object can be in two possible states: good or bad. We further assume that the actual object state is not known at allocation time, but that the decision-maker knows the probability for each object to be in each state. We propose a formal model of this problem, based on the notions of *ex-ante* and *ex-post* fairness, and we propose some algorithms aiming at computing optimal allocations in the sense of *ex-post* egalitarianism, the efficiency of these algorithms being tested on random instances.

1 Introduction

The problem of allocating a set of indivisible goods to a set of agents arises in a wide range of applications including, among others, auctions, divorce settlements, frequency allocation, airport traffic management, fair and efficient exploitation of Earth observing satellites [8]. In many such real-world problems, one needs to find *fair* solutions where fairness refers to the need for compromises between the agents (often antagonistic) objectives.

While most works (see *e.g.* [4] for a survey on multiagent resource allocation) on fair division typically assume that the agents are able to evaluate their preferences (ranking, utility function) over the sets of objects at stake before the beginning of the allocation process, it might not always be the case, and the actual value (or state) of some given objects may depend on exogenous factors and not be known by the agents beforehand. This is the case for example in the fair share of a constellation of Earth observing satellites [8], as the weather conditions on a given area, which are only known with a given probability when the allocation is decided, can dramatically reduce the quality of the observation, and, in the end the utility of an observation for an agent.

Uncertainty (or, more precisely, risk) issues in collective decision making have been studied for example by Myerson [10] and more recently Gajdos and Tallon [5]. However, to the best of our knowledge, this problem has never been considered from a computational point of view, except within the combinatorial auctions framework, when one wants to minimize the influence in terms of revenue of potential bids withdrawals [7]. Our work aims at bridging this gap.

In this article, we make three main assumptions. (i) The allocation is *centralized*, that is, it is decided and computed by a central benevolent authority, according to the agents' individual preferences. (ii) Each object can only be in two possible conditions (good or bad). The actual condition of each object is only known with a given probability when the allocation is decided, but is known for sure when the objects are actually allocated to the agents. (iii) The agents have non exogenous additive preferences over the objects. In other words, the preferences of each agent are represented by a set of weights, standing for the utility (or satisfaction) she enjoys for each single object. The utility of an agent for a subset of objects S is then given by the sum of the weights of all the objects in S that are in good condition (we assume that a bad object has absolutely no value for the agent who receives it).

Even if this framework seems restrictive, we advocate that it is worth studying for the following reasons. Firstly, the additivity assumption is very natural as soon as preferences over sets of objects have to be represented in a compact way. Secondly, in many real-world problems, uncertainty can be defined "object-wise" and thus can be very naturally modeled as we suggest. Finally, as we show in this paper, despite its apparent simplicity, our framework raises non trivial computational issues.

This article is structured as follow. In Section 2, we introduce our framework for fair division of indivisible goods under risk. In Section 3, we mainly focus on the computation of optimal or good ex-post egalitarian allocations and we propose two algorithms to solve this problem. Finally, we compare the efficiency of these algorithms on random instances in Section 4.

2 Framework

2.1 Model

In the following, we use lower case bold font to represent vectors and upper case bold font to represent matrices.

A finite set of indivisible *objects* $\mathcal{O} = \{1, \ldots, l\}$ must be allocated to a finite set of *agents* $\mathcal{A} = \{1, \ldots, n\}$. An *allocation decision* (or simply *allocation*) is a vector of shares $\boldsymbol{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ where $\pi_i \subseteq \mathcal{O}$, and $j \in \pi_i$ iff object j

has been given to agent i. The set of feasible decisions is $\mathcal{D} = \{\boldsymbol{\pi},\ i \neq i' \Rightarrow \pi_i \cap \pi'_i = \emptyset\}$. We further denote by $\pi_0 = \mathcal{O} \setminus \bigcup_{i \in \mathcal{A}} \pi_i$ the set of non allocated objects.

Each object can be either in *good* condition or in *bad* condition. The objects conditions are known only after the allocation has been made, but the decision-maker is nevertheless given probabilistic information: to each object $j \in \mathcal{O}$, is attached a binary random variable X_j which can take value in $\{good, bad\}$. We assume the existence of a vector $\mathbf{p} \in [0;1]^l$ giving each object probabilities $p_j = \mathbb{P}(X_j = good)$, and $\overline{p}_j = 1 - p_j = \mathbb{P}(X_j = bad)$. Variables $X_j,\ j \in \mathcal{O}$ are assumed to be independent.

Each state of nature in the problem is therefore characterized by the set of objects in good condition (the other ones being in bad condition). Let $\mathcal{S} = \{1, \ldots, k\}$ be the set of the possible *states of nature* (where $k = 2^l$) ; to each state s of this set, one can relate the set $good(s) \subseteq \mathcal{O}$ of objects in good condition when state of nature s happens. $\mathcal{S}$ is provided with a probability distribution, fully characterized by coefficients p_j.

$$\forall s \in \mathcal{S},\ \Pr(s) = \prod_{j \in good(s)} p_j \prod_{j \notin good(s)} \overline{p}_j \qquad (1)$$

Computing an acceptable allocation for such a problem requires the decision-maker to know about the tastes of the agents for the objects. These *preferences* are numerically expressed by the agents in the form of *utility functions*, which, for each state s, map each decision $\boldsymbol{\pi}$ to a numerical value $u_{i,s}(\boldsymbol{\pi})$ conveying the attractiveness of the decision for the agent i if this state of nature happens. This utility is built upon the specification of *weights* for each agent to each object ; the weight w_{ij} represents the intensity of agent i's preference for object j ; we assume that an agent utility for a decision and a state of nature are given by the sum of the weights of the objects *in good condition* received by said agent: agents have *additive* preferences over the objects, and each object in bad condition gives no extra utility to the agent it is allocated to.

$$\forall i \in \mathcal{A}, \forall s \in \mathcal{S},\ u_{i,s}(\boldsymbol{\pi}) = \sum_{j \in good(s) \cap \pi_i} w_{ij} \qquad (2)$$

Let us now define an instance of the problem studied in this article.

Definition 1 (Resource allocation problem under risk) *An instance of a resource allocation problem under risk is a tuple* $(\mathcal{A}, \mathcal{O}, \mathbf{p}, \mathbf{W})$*, where* $\mathcal{A} = \{1, \ldots, n\}$ *is a set of agents,* $\mathcal{O} = \{1, \ldots, l\}$ *is a set of objects,* $\mathbf{p} \in [0;1]^l$ *expresses the probability for each object to be in good condition, and* $\mathbf{W}$ *is the* n*-lines* l*-columns matrix of weights given to the objects by the agents.*

Table 1 shows an example of a resource allocation problem under risk, with the probabilities of each possible state of nature (line 2) and utility profiles associated with a given decision (lines 3 and 4).

2.2 The *timing effect*

For a given state of nature, a decision quality depends on the level of satisfaction of all the agents. A classical way to define this quality is to *aggregate* the agents utility vector with a commutative and increasing *collective utility function* $\mathfrak{M} : (\mathbb{R}^+)^n \to \mathbb{R}^+$, which measures social welfare. Two classical choices are $\mathfrak{M} = \sum$ and $\mathfrak{M} = \min$, which have been at the root of classical utilitarianism on the one hand, and egalitarianism on the other hand. The latter promotes equity, since best decisions are those which satisfy the most the poorest agent, whereas the former promotes a kind of efficiency which aims at giving objects to the agents producing the most utility, without any concern for equity. A general survey on collective utility functions can be found in [9]. In the following we will write $\mathfrak{M}_{i \in \mathcal{A}} u_i$ for $\mathfrak{M}(\mathbf{u})$.

In the same manner, we aggregate agent utilities in the different states of nature using the classical expected utility (even if other choices could be made).

In order to map a unique numerical value to each decision, and depending on whether aggregation is first made over states of nature and then over agents or the other way around, we obtain two different functions [6; 10] : $acu : \mathcal{D} \to \mathbb{R}^+$, defined in (3), is called *ex-ante* collective utility and $pcu : \mathcal{D} \to \mathbb{R}^+$, defined in (4) is called *ex-post* collective utility.

$$\forall \boldsymbol{\pi} \in \mathcal{D},\ acu(\boldsymbol{\pi}) = \mathop{\mathfrak{M}}_{i \in \mathcal{A}} \left(\sum_{s \in \mathcal{S}} \Pr(s) \cdot u_{i,s}(\boldsymbol{\pi}) \right) \qquad (3)$$

$$\forall \boldsymbol{\pi} \in \mathcal{D},\ pcu(\boldsymbol{\pi}) = \sum_{s \in \mathcal{S}} \Pr(s) \cdot \left(\mathop{\mathfrak{M}}_{i \in \mathcal{A}} u_{i,s}(\boldsymbol{\pi}) \right) \qquad (4)$$

Harsanyi [6] shows that the only aggregation functions for which *ex-post* and *ex-ante* utilities coincide are linear or affine, which entails that, on the contrary, each equity-prone collective aggregation function will give different *ex-ante* and *ex-post* utilities. There therefore exists a conflict – known as *timing effect* – between the *ex-post* approach on the one hand, which considers the expected social welfare and the *ex-ante* approach on the other hand, which considers the social welfare measured with expected utilities.

2.3 *Ex-ante* versus *ex-post* utility

Even if no link exists *a priori* between *ex-post* and *ex-ante* utilities for a given decision, one can show that, under some mild assumption on the collective aggregation function, the *ex-ante* collective utility is always greater than the *ex-post* one.

This is especially true in the egalitarian case, where Proposition 1 is a direct application of the triangular inequality for function $\min$.

Proposition 1 *Let* $\mathfrak{M} = \min$ *be the egalitarian collective aggregation operator. Then, the following inequality stands:*

$$\forall \boldsymbol{\pi} \in \mathcal{D},\ pcu(\boldsymbol{\pi}) \leq acu(\boldsymbol{\pi}) \qquad (5)$$

3 Computing *ex-ante* and *ex-post* optimal allocations

In this section, we will deal with the problems of finding an allocation maximizing *ex-ante* and *ex-post* utilities. In the following, we will restrict to the classical egalitarian criterion –

s	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\dots$	$\{2,3,4\}$	$\{1,2,3,4\}$	$\mathfrak{E}(\mathbf{u})$
$\Pr(s)$	0.016	0.004	0.016	0.004	0.064	0.016	0.064	$\dots$	0.256	0.064	—
$u_{1,s}$	0	10	0	0	7	10	10	$\dots$	7	17	9.4
$u_{2,s}$	0	0	8	4	0	8	4	$\dots$	12	12	8.4
$\mathfrak{M}(\mathbf{u})$	0	0	0	0	0	8	4	$\dots$	7	12	8.4 / 6.448

Table 1: Utility profile and *ex-ante* and *ex-post* utility computation for a problem with 2 agents, 4 objects, probabilities $\mathbf{p} = \langle 0.8, 0.5, 0.5, 0.2\rangle$, weights $\mathbf{w_1} = \langle 10, 2, 4, 7\rangle$ and $\mathbf{w_2} = \langle 3, 8, 4, 10\rangle$, decision $\boldsymbol{\pi} = \langle\{1,4\},\{2,3\}\rangle$, $\mathfrak{M} = \min$. Here, $pcu(\boldsymbol{\pi}) = 6.448$ and $acu(\boldsymbol{\pi}) = 8.4$ (see Section 2.2).

that is, $\mathfrak{M} = \min$ – which is worthy of attention in this context, as it represents exactly the expected utility of the poorest agent.

Ex-ante collective utility *Ex-ante* collective utility is defined by Equation (3) ; introducing some "expected weights" $\tilde{w}_{ij} = p_j w_{ij}$, the expression can be simplified: $\forall \boldsymbol{\pi} \in \mathcal{D}$, $acu(\boldsymbol{\pi}) = \mathfrak{M}_{i\in\mathcal{A}}\ \tilde{u}_i(\boldsymbol{\pi})$ where $\tilde{u}_i(\boldsymbol{\pi}) = \sum_{j\in\pi_i} \tilde{w}_{ij}$.

Thus, since the $\tilde{w}_{ij}$ coefficients can be computed in mere linear-time, the problem of finding an *ex-ante* optimal allocation can be reduced to a classical risk-free resource allocation problem with additive preferences, known as the Santa Claus problem [1]. Since this problem has already been tackled in litterature, we focus in the following on the *ex-post* optimization problem.

Ex-post collective utility A basic algorithm for *computing* the *ex-post* collective utility, directly applying formula (4), requires the computation of the collective utility in each possible state (*i.e* each column in Table 1), that is, the enumeration of an exponential number of values. Clearly, computing the *ex-post* collective utility of a given decision is in $\#\mathsf{P}$, but we do not know yet if it is complete for this class (even if we strongly believe it).[1]

However, as soon as all the objects allocated to an agent are in bad condition, the utility of this agent is zero, and so is the collective utility, whatever states the remaining objects are in. Algorithm 1, which computes the *ex-post* collective utility for a given decision, is based on this remark: it quickly "eliminates" such states of nature, whose enumeration is unnecessary. A function SORT is used in this algorithm in the following manner: $\text{SORT}(\mathbf{u}, f)$ returns a vector $\mathbf{u}^{\uparrow}$ which is a permutation of the values of $\mathbf{u}$, such that $i < i' \Rightarrow f(u_i^{\uparrow}) \leq f(u_{i'}^{\uparrow})$.

The *optimization* problem is tackled with both exact and approximate algorithms.

The exact approach is based on a classic *branch and bound* algorithm. Efficiency of such an algorithm highly depends on its ability to quickly detect poor allocations in order to "cut" significant parts of the search tree. A cut must be based on an easy-to-compute function which maximizes the value to be optimized.

Ex-post utility computation is time-consuming, and is therefore not used as a cut strategy, but only to assess complete allocations.

[1] Of course, computing an optimal allocation is even harder.

Algorithm 1: EXPOST function: *ex-post* collective utility computation

Data: A complete allocation $\boldsymbol{\pi}$
Result: *Ex-post* collective utility $pcu(\boldsymbol{\pi})$

$\boldsymbol{\pi}^{\uparrow} \leftarrow \text{SORT}(\langle\pi_1, \dots, \pi_n\rangle, \mathcal{X} \mapsto |\mathcal{X}|)$;
return BRANCH($\langle 0, \dots, 0\rangle$, 1, $\boldsymbol{\pi}^{\uparrow}$, 1);

Function BRANCH($\mathbf{u}$, pr, $\langle\rho_1, \dots, \rho_n\rangle$, i)

Data: A utility vector $\mathbf{u}$, a number $pr \in [0;1]$, a vector of shares $\boldsymbol{\rho}$, an agent i
Result: *Ex-post* collective utility

if $\rho_i = \emptyset$ **then**
  if $i = n$ **then**
    return $\min(\mathbf{u}) \times pr$;
  else
    if $u_i = 0$ **then return** 0;
    return BRANCH($\mathbf{u}$, pr, $\boldsymbol{\rho}$, $i+1$);
else
  $j \leftarrow$ arbitrary object in ρ_a;
  $\boldsymbol{\rho'} \leftarrow \langle\dots, \rho_{i-1}, \rho_i \setminus \{j\}, \rho_{i+1}, \dots\rangle$;
  $\mathbf{u'} \leftarrow \langle\dots, u_{i-1}, u_i + w_{ij}, u_{i+1}, \dots\rangle$;
  return BRANCH($\mathbf{u}$, $pr \cdot \overline{p}_j$, $\boldsymbol{\rho'}$, i) + BRANCH($\mathbf{u'}$, $pr \cdot p_j$, $\boldsymbol{\rho'}$, i);

Instead, we use inequality (5) and choose function $\overline{acu}$ as upper bound ; $\overline{acu}$ represents the *ex-ante* utility of a virtual decision which would allocate to *all the agents* the set of objects (denoted π_0) that are not yet allocated by the current decision $\boldsymbol{\pi}$:

$$\overline{acu}(\boldsymbol{\pi}) = \min_{i\in\mathcal{A}}\Big(\sum_{j\in\pi_i}\tilde{w}_{ij} + \sum_{j\in\pi_0}\tilde{w}_{ij}\Big)$$

Even though $\overline{acu}$ is clearly a rough upper bound, this value remains fast to compute.

At this point, it seemed interesting to look for a intermediate function, which would be a better upper bound than $\overline{acu}$ and faster to compute than pcu. The idea is to compute utility in an *ex-post* manner for a subset Ω of objects, and in an *ex-ante* manner for the other ones ; we introduce in this sense the *mixed utility*, denoted $mu_{i,s}$ for a given agent i and a given state of nature s.

Algorithm 2: Stochastic greedy

Data: A risky fair division problem instance.
Result: A good allocation, according to *ex-post* collective utility

$Stock \leftarrow \emptyset$;
$\boldsymbol{\pi}^\star = \langle \pi_1^\star, .., \pi_l^\star \rangle \leftarrow \langle \emptyset, \ldots, \emptyset \rangle$;
$pcu^\star \leftarrow 0$;
$i \leftarrow 0$;
while *given time has not elapsed* **do**
  $\boldsymbol{\pi} \leftarrow$ BUILDALLOCATION() ;
  if $acu(\boldsymbol{\pi}) \geq pcu^\star$ **then**
    $pcu_{app} \leftarrow$ EXPOSTA($\boldsymbol{\pi}$) ;
    if $pcu_{app} > \min_{\boldsymbol{\pi} \in Stock}(\text{EXPOSTA}(\boldsymbol{\pi}))$ **then**
      STORE($\boldsymbol{\pi}$) ;
  $i \leftarrow i + 1$;
  if $i = nbStorage \times nbBeforeExactComputation$ **then**
    for $\boldsymbol{\pi} \in Stock$ **do**
      $pcu \leftarrow$ EXPOST($\boldsymbol{\pi}$) ;
      if $pcu > pcu^\star$ **then**
        $\boldsymbol{\pi}^\star \leftarrow \boldsymbol{\pi}$;
        $pcu^\star \leftarrow pcu$;
    $Stock \leftarrow \emptyset$;
    $i \leftarrow 0$;
return $\boldsymbol{\pi}^\star$;

Procedure BUILDALLOCATION()

$\mathbf{u} = \langle u_1, .., u_n \rangle \leftarrow \langle 0, \ldots, 0 \rangle$;
$\boldsymbol{\pi} = \langle \pi_1, .., \pi_n \rangle \leftarrow \langle \emptyset, \ldots, \emptyset \rangle$;
while $\exists j \in \pi_0$ **do**
  $\mathring{i} \leftarrow \text{argmin}_{i \in \mathcal{A}}(alter(u_i))$;
  $\mathring{j} \leftarrow \text{argmax}_{j \in \pi_0}(alter(w_{\mathring{i}j}))$;
  $\pi_{\mathring{i}} \leftarrow \pi_{\mathring{i}} \cup \mathring{j}$;
  $u_{\mathring{i}} \leftarrow u_{\mathring{i}} + w_{\mathring{i}\mathring{j}}$;
return $\boldsymbol{\pi}$;

$$mu_{i,s}(\boldsymbol{\pi}, \Omega) = \sum_{\substack{j \in \Omega \cap good(s) \\ j \in \pi_i}} w_{ij} + \sum_{\substack{j \notin \Omega \\ j \in \pi_i}} \tilde{w}_{ij}$$

The mixed utility represents the utility of an agent which considers that objects outside Ω are for sure in good condition and which assigns them weights $\tilde{w}_{ij}$. The *mixed collective utility* is defined by Equation (6) as the *ex-post* collective utility from individual mixed utilities.

$$mcu(\boldsymbol{\pi}, \Omega) = \sum_{s \in \mathcal{S}} \Pr(s) \cdot \min_{i \in \mathcal{A}} mu_{i,s}(\boldsymbol{\pi}, \Omega) \quad (6)$$

Note that individual mixed utilities are independent from the states of the objects outside Ω. The expected value computation in Equation (6) can therefore boil down to the formula (7), where for s s.t. $good(s) \subseteq \Omega$, one denotes $\Pr(s, \Omega) = \prod_{j \in good(s)} p_j \prod_{j \in \Omega \setminus good(s)} \overline{p}_j$ the probability for objects in Ω to be in the state specified by s, whatever states the other objects are in. The number of states of nature to list is halved for each object outside Ω, which shows the algorithmic point of mixed collective utility.

$$mcu(\boldsymbol{\pi}, \Omega) = \sum_{\substack{s \in \mathcal{S} \\ good(s) \subseteq \Omega}} \Pr(s, \Omega) \cdot \min_{i \in \mathcal{A}} um_{i,s}(\boldsymbol{\pi}, \Omega) \quad (7)$$

We can prove that mixed collective utility lies between *ex-post* and *ex-ante* collective utilities (proof omitted due to lack of space).

Proposition 2 (Mixed collective utility) *For all decision $\boldsymbol{\pi} \in \mathcal{D}$ and for all subset $\Omega \subseteq \mathcal{O}$, one has:*

$$acu(\boldsymbol{\pi}) \geq mcu(\boldsymbol{\pi}, \Omega) \geq pcu(\boldsymbol{\pi}) \quad (8)$$

Our *branch and bound* algorithm uses the upper bound function $\overline{acu}$ to cut within the body of research: the function mcu is used only when a complete allocation has been made, to avoid unnecessary *ex-post* collective utility computations.

Dynamic heuristics are used as suggested by [2] : each object will be firstly allocated to the poorest agent (i.e. the one whose expected utility is currently the lower) ; when a new object has to be allocated, the one preferred by the currently poorest agent is chosen among those still left.

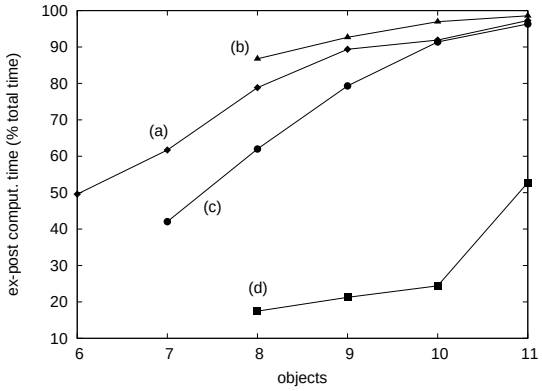
The approximate algorithm (Algorithm 2) is based upon a greedy stochastic algorithm [3]. As soon as a complete allocation has been built, an approximate *ex-post* collective utility computation is made by EXPOSTA, in order to decide if the allocation will be stored or not. The approximate computation is made using the mixed collective utility or the Monte-Carlo method (the latter being based on a sequence of random draws in the space of states of nature). A fixed number $nbStorage$ of promising allocations is stored within the course of the algorithm ; if an allocation is better – as far as the approximate computation can tell – than the worst currently stored, the function STORE saves this new allocation (and the other one is deleted if the storage capacity is reached). As soon as $nbStorage \times nbBeforeExactComputation$ allocations have been made, an exact *ex-post* collective utility computation occurs for each stored allocation, and only the best one is kept.

During the building of an allocation, we use randomly biased heuristics, introducing function $alter : \mathbb{R} \to \mathbb{R}$, such that $\forall y \in \mathbb{R},\ alter(y) = y \cdot (1 + \phi X)$, where ϕ is a positive real parameter and X a standard normal random variable.

4 Results

Algorithms introduced in this article are implemented using Java and run on random instances, where weights w_{ao} are uniformly drawn in $\{0, 1, .., 99\}$, and probabilities p_j uniformly in $[0; 1]$.

Table 2 and Figure 1 show the results of the exact search algorithm. Four configurations are tested: the algorithm is firstly run with a cut based upon $\overline{acu}$ function only (case (a)), then by using dynamic heuristics (case (b)), next by introducing Algorithm 1 for *ex-post* collective utility computation (case (c)), and finally by adding mixed collective utility cuts (case (d)). Figure 1 shows efficiency of configuration (d),



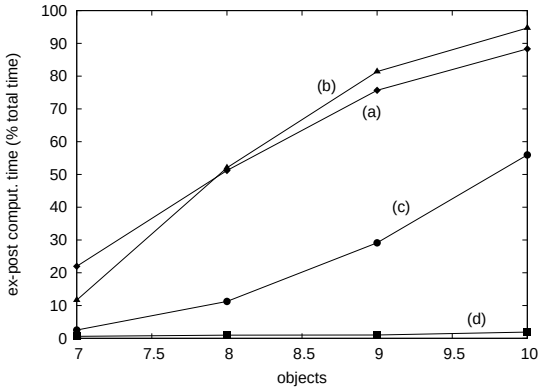


Figure 1: Exact resolution. Duration of *ex-post* collective utility computations, as a percentage of total execution time, for 5 (left) and 7 (right) agents (mean over 100 instances)

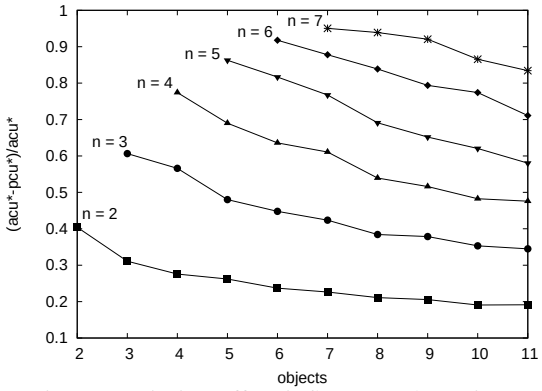


Figure 2: Timing effect influence. The ratio $(acu^\star - pcu^\star)/acu^\star$ varies with the number of objects, for different numbers of agents..

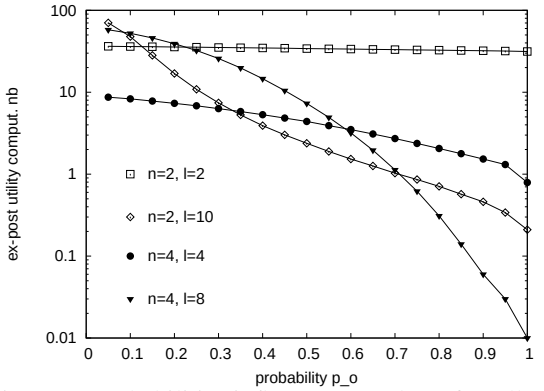
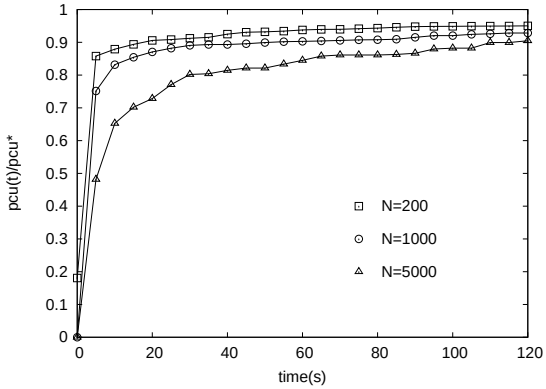
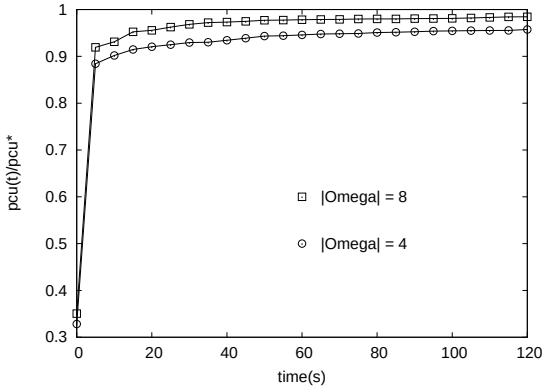


Figure 3: Probabilities influence. Number of totally explored search tree branches, as a percentage of the total number of branches (mean over 100 instances for different numbers of agents and "equaly likely" objects)



(a) Monte-Carlo approximation, for different numbers of draws.



(b) Mixed collective utility approximation, for different sizes of the Ω set.

Figure 4: Approached resolution. Evolution of the best *ex-post* collective utility with time, for two approximation methods (means over 100 instances involving 5 agents and 12 objects).

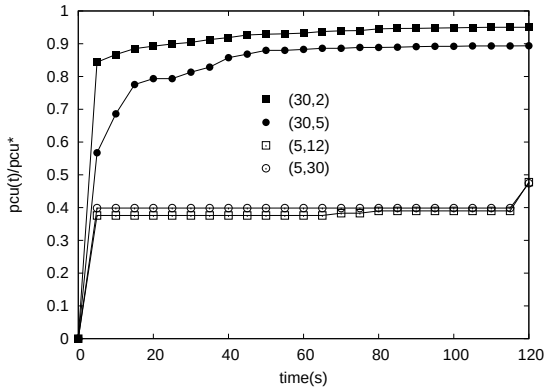


Figure 5: Approached resolution with Monte-Carlo method. *Ex-post* collective utility evolution for different couples $(nbStorage, nbBeforeExactComputation)$ (means over 100 instances involving 5 agents and 12 objects)

in which use of mixed collective utility produces good cuts, and therefore deeply reduces the number of *ex-post* collective utility computations made during the algorithm.

n	l	(a)	(b)	(c)	(d)
5	≤ 9	100	100	100	100
5	10	49	52	89	100
5	11	1	1	10	52
5	≥ 12	0	0	0	0
7	≤ 8	100	100	100	100
7	9	27	47	100	100
7	10	0	1	19	32
7	≥ 11	0	0	0	0

Table 2: Exact resolution. Number of instances solved in 30 seconds (over 100 random instances).

Figure 2 shows influence of the *timing effect*. Relative difference between *ex-post* and *ex-ante* collective utilities increases when the number of agents increases or when the number of objects decreases.

The algorithm efficiency highly depends on probabilities $\mathbf{p}$, which is clearly illustrated by Figure 3. Because of higher proximity between *ex-ante* and *ex-post* collective utilities when the probabilities p_j are closer to 1, cutting strategies are more efficient in this case.

Algorithm 2 is tested on 100 instances ($n = 5$, $l = 12$), for a duration of 2 minutes[2]. Figure 4 illustrates the influence of the approximation methods parameters ; Figure 5 shows the importance of functionnal parameters: the best solution quality significantly increases with the number of allocations stored during the run.

5 Conclusion

In this article, we have introduced a simple model for resource allocation problems under risk. We have shown that, under reasonable hypothesis, *ex-ante* collective utility optimization could be reduced to risk-free optimization, but that *ex-post* optimization seemed to be far more complex. We have proposed the mixed collective utility as groundwork for the building of both an exact and an approximate algorithm.

Algorithms introduced in this article are a first attempt at solving risky resource allocation problems and can most probably be improved. Further work has to be made to characterize the complexity of the *ex-post*-related problems. Moreover, the *ex-post* egalitarian framework shows its limits when $l \leq n$ due to the drawning effect induced by function $\min$. We plan next to extend the model, in order to work with other collective utility aggregations such as the leximin ordering, consider preferential and/or probabilistic dependences between objects, and to embrace a more general notion of risk.

Acknowledgements We wish to thank Jérôme Lang and Dídac Busquets for their numerous remarks during the genesis of the present article. Authors would also like to thank the anonymous reviewers for their comments and suggestions.

References

[1] N. Bansal and M. Sviridenko. The santa claus problem. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 31–40. ACM, 2006.

[2] S. Bouveret and M. Lemaître. Computing leximin-optimal solutions in constraint networks. *Artificial Intelligence*, 173(2):343–364, 2009.

[3] J. L. Bresina. Heuristic-Biased Stochastic Sampling. In *AAAI-96*, pages 271–278, Portland, OR, 1996.

[4] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[5] T. Gajdos and J.-M. Tallon. Fairness under uncertainty. *Economics Bulletin*, 4(18):1–7, 2002.

[6] J. C. Harsanyi. Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility. *Journal of political economy*, 63:309–321, 1955.

[7] A. Holland and B. O'Sullivan. Robust solutions for combinatorial auctions. In *Proc. of the 6th ACM Conf. on Electronic Commerce*, pages 183–192. ACM, 2005.

[8] M. Lemaître, G. Verfaillie, and N. Bataille. Exploiting a common property resource under a fairness constraint: a case study. In *IJCAI-99*, pages 206–211, Stockholm, Sweden, July 1999.

[9] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[10] R. B. Myerson. Utilitarianism, egalitarianism, and the timing effect in social choice problems. *Econometrica*, 49(4):883–897, 1981.

[2]Exact resolution of problems of this size takes 5 to 10 minutes.

Influencing and aggregating agents' preferences over combinatorial domains

Nicolas Maudet[1], **Maria Silvia Pini**[2], **Francesca Rossi**[2], **K. Brent Venable**[2]
1: LAMSADE, Univ. Paris Dauphine, France
Email: Nicolas.MAUDET@dauphine.fr
2: Department of Pure and Applied Mathematics,
University of Padova, Italy
Email: {mpini,frossi,kvenable}@math.unipd.it

Abstract

In a multi-agent context where a set of agents declares their preferences over a common set of candidates, it is often the case that such agents interact and exchange opinions before voting. In this initial phase, agents may influence each other and therefore modify their preferences, until hopefully they reach a stable state. Recent work has modelled the influence phenomenon in the case of voting over a single issue. Here we generalize this model to account for preferences over combinatorially structured domains including several issues. When agents express their preferences as CP-nets, we show how to model influence functions and to aggregate preferences by possibly interleaving voting and influence convergence.

1 Introduction

In a multi-agent context where a set of agents declares their preferences over a common set of candidates, it is often the case that such agents interact and exchange opinions before voting. For example, in political elections, polls provide a representative sample of the opinion of the voters, and some influential people may declare their vote inclination. Moreover, in social networks, people often exchange their opinions before taking a decision.

In this initial phase, agents may influence each other and therefore modify their preferences. For example, in political elections, a voter may be influenced by the opinion of esteemed people. In a work environment, the participants to a project meeting may have to take one or more decisions about the project plan and may be influenced by the opinion of experts of the field.

The concept of influence has been widely studied in psychology, economics, sociology, and mathematics [DeGroot, 1974; P. DeMarzo, 2003; Krause, 2000]. Recent work has modelled the influence phenomenon in the case of taking a decision over a single issue [Grabisch and Rusinowska, 2010]. In this influence framework, each agent has two possible actions to take and it has an inclination to choose one of the actions. Due to influence by other agents, the decision of the agent may be different from the original inclination. The transformation from the agent's inclination to its decision is represented by an influence function. It is also interesting to draw connection to the recent work on (some kind of) manipulation in computational social choice. In so-called *bribery* problems [Faliszewski *et al.*, 2009], an agent has typically a limited budget he can spend to modify the vote of other agents. In the *safe manipulation* setting [Slinko and White, 2008], it is assumed that an influential agent can be imitated in his vote by a proportion of followers. These are clearly specific notions of influence, but restricted in the sense that a single influencing agent is considered, and that the process is simply one-shot. In many real scenarios, influence among agents does not stop after one step but it is an iterative process.

Here we generalize these models to account for preferences over combinatorially structured domains including several issues. In fact, often a set of agents needs to select a common decision from a set of possible decisions, over which they express their preferences, and such a decision set has a combinatorial structure, that is, it can be seen as the combination of certain issues, where each issue has a set of possible instances. Consider for example a car: usually it is not seen as a single item, but as a combination of features, such as its engine, its shape, its color, and its cost. Each of these features has some possible instances, and a car is the combination of such feature instances. If a family needs to buy a new car, each family member may have his own opinion about cars, and the task is to choose the car that best fits the preferences of everybody.

Usually preferences over combinatorially structured domains are expressed compactly, otherwise too much space would be needed to rank all possible alternatives. CP-nets are a successful framework that allows one to do this [Boutilier *et al.*, 2004]. They exploit the independence among some features to give conditional preferences over small subsets of them.

CP-nets have already been considered in a multi-agent setting [Rossi *et al.*, 2004; Lang and Xia, 2009; Purrington and Durfee, 2007; Xia *et al.*, 2008]. Here we adapt such frameworks to incorporate influences among agents. We allow influences to be over the same issue or also among different issues. We show how to model influence functions and we observe that influence and conditional preferential dependency in CP-nets have the same semantic model. This allows us to naturally embed influences in a multi-agent CP-net profile.

We then propose a way to aggregate preferences by possibly interleaving voting and influence convergence.

2 Background

2.1 Influence functions

In [Grabisch and Rusinowska, 2010] a framework to model influences among agents in a social network environment is defined. Each agent has two possible actions to take and it has an inclination to choose one of the actions. Due to influence by other agents, the decision of the agent may be different from its original inclination. The transformation from the agent's inclination to its decision is represented by an influence function. In many real scenarios, influence among agents does not stop after one step but it is an iterative process.

Any influence function over n agents can be modelled via a matrix with 2^n rows and 2^n columns, where each row and column correspond to a certain state (a vector containing the agents' inclinations). A 1 in the cell (S, T) of the matrix means that from state S we pass to state T via the influence function. Alternatively, the influence function can be modelled via a graph where nodes are states and arcs model state transitions via the influence function. If we adopt the iterative model of influence, we may pass from state to state until stability holds (that is, in the graph formulation, we are in a state represented by a node with a loop), or we may also not converge.

Let us consider some examples of influence functions, as defined in [Grabisch and Rusinowska, 2010]:

- The **Fol** influence function considers two agents, each of which follows the inclination of the other one. This influence function converges to stability only when the initial inclination models consensus between the two agents. If we start from another state, influence iteration never stops.
- On the other hand, in the **Id** influence function, where each of agent follows only its own inclination, all states are stable.
- Another example is the influence function modelling the presence of a guru, called **Gur**, where one of the agents is the guru and all other agents follow him. Such a function has two states, which both represent consensus. Given any initial inclination, the iteration will converge to one of the stable states.
- A final example, that we will consider also later in the paper, is the **Conf3** influence function, that models a community with 4 people which follow a Confucian model. The four people are a king, a man, a woman, and a child. The man follows the king, the woman and child follow the man, and the king is influenced by others only if he has a positive inclination, in which case he will follow such an inclination only if at least one of the other people agrees with him. In [Grabisch and Rusinowska, 2010] it is shown that this influence function always converges to one of two stable states, which both represent consensus, depending on the initial state.

2.2 CP-nets

CP-nets [Boutilier *et al.*, 2004] are a graphical model for compactly representing conditional and qualitative preference relations. CP-nets are sets of *ceteris paribus (cp)* preference statements. For instance, the statement "*I prefer red wine to white wine if meat is served.*" asserts that, given two meals that differ *only* in the kind of wine served *and* both containing meat, the meal with red wine is preferable to the meal with white wine.

Formally, a CP-net has a set of features $F = \{x_1, \ldots, x_n\}$ with finite domains $\mathcal{D}(x_1), \ldots, \mathcal{D}(x_n)$. For each feature x_i, we are given a set of *parent* features $Pa(x_i)$ that can affect the preferences over the values of x_i. This defines a *dependency graph* in which each node x_i has $Pa(x_i)$ as its immediate predecessors. Given this structural information, the agent explicitly specifies her preference over the values of x_i for *each complete assignment* on $Pa(x_i)$. This preference is assumed to take the form of total or partial order over $\mathcal{D}(x_i)$. An *acyclic* CP-net is one in which the dependency graph is acyclic.

Consider a CP-net whose features are A, B, C, and D, with binary domains containing f and $\overline{f}$ if F is the name of the feature, and with the preference statements as follows: $a \succ \overline{a}$, $b \succ \overline{b}$, $(a \wedge b) \vee (\overline{a} \wedge \overline{b}) : c \succ \overline{c}$, $(a \wedge \overline{b}) \vee (\overline{a} \wedge b) : \overline{c} \succ c$, $c : d \succ \overline{d}$, $\overline{c} : \overline{d} \succ d$. Here, statement $a \succ \overline{a}$ represents the unconditional preference for $A = a$ over $A = \overline{a}$, while statement $c : d \succ \overline{d}$ states that $D = d$ is preferred to D=$\overline{d}$, given that $C = c$.

The semantics of CP-nets depends on the notion of a worsening flip. A *worsening flip* is a change in the value of a variable to a less preferred value according to the cp-statement for that variable. For example, in the CP-net above, passing from $abcd$ to $ab\overline{c}d$ is a worsening flip since c is better than $\overline{c}$ given a and b. One outcome α is *better* than another outcome β (written $\alpha \succ \beta$) iff there is a chain of worsening flips from α to β. This definition induces a preorder over the outcomes, which is a partial order if the CP-net is acyclic.

In general, finding the optimal outcome of a CP-net is NP-hard [Boutilier *et al.*, 2004]. However, in acyclic CP-nets, there is only one optimal outcome and this can be found in linear time by sweeping through the CP-net, assigning the most preferred values in the preference tables. For instance, in the CP-net above, we would choose $A = a$ and $B = b$, then $C = c$, and then $D = d$. In the general case the optimal outcomes coincide with the solutions of a constraint problem obtained replacing each cp-statement with a constraint [Brafman and Dimopoulos, 2004]. For example, the following cp-statement (of the example above) $(a \wedge b) \vee (\overline{a} \wedge \overline{b}) : c \succ \overline{c}$ would be replaced by the constraint $(a \wedge b) \vee (\overline{a} \wedge \overline{b}) \Rightarrow c$.

In the context of preference aggregation, CP-nets have been used as a compact way to represent the preferences of each voter. In particular, in [Lang and Xia, 2009] the authors showed that a sequential single-feature voting protocol can find a winner object in polynomial time. Moreover, such an approach has several other desirable properties, when the CP-nets satisfy a certain condition on their dependencies called *O-legality*. In [Lang and Xia, 2009], the CP-nets must be acyclic, and their dependency graphs must all be compatible

with a given graph ordered according to the feature ordering in the voting procedure. In other words, there is a linear order O over the features such that for each voter the preference over a feature is independent of features following it in O.

3 Modelling influence

The setting we consider consists of a set of n agents expressing their preferences over a common set of candidates. The candidate set has a combinatorial structure: there is a common set of features and the set of candidates is the Cartesian product of their domains. Thus each candidate is an assignment of values to all features.

For the sake of simplicity of the technical developments of this paper, we assume features to be binary (that is, with two values in their domain). However, the approach we propose can be generalized to non-binary features.

Each agent expresses its preferences over the candidates via an acyclic CP-net. Moreover, we assume that these CP-nets are compatible: given n CP-nets $N_1, \ldots, N_n$, they are said to be compatible if the union of their dependency graphs, that we call $Dep(N_1, \ldots, N_n)$, does not contain cycles. Notice that compatible CP-nets do not necessarily have the same dependency graph.

Definition 1 *Given n agents and m binary features, a profile is a collection of n compatible CP-nets over the m features.*

We note that our notion of profile coincides with the notion of O-legal profile in [Lang and Xia, 2009].

Given a profile P with CP-nets $N_1, \ldots, N_n$, we will abuse the notation and often write $Dep(P)$ to mean $Dep(N_1, \ldots, N_n)$.

A profile models the initial inclination of all agents, that is, their opinions over the candidates before they are influenced by each other.

Since the set of features is the same for all agents, but each agent may have a possibly different CP-net, to avoid confusion we call variables the binary entities of each CP-net. Thus, given a profile with m features, for each feature there are n variables modelling such a feature, one for each CP-net. Thus the whole profile has $m * n$ variables.

3.1 Conditional influence

A straightforward way to include influences into profiles is to have influence functions act on each single feature, as in [Grabisch and Rusinowska, 2010]. That is, the preferences of an agent over a certain feature may be influenced by the preferences of one or more other agents over the same feature.

While influence functions in [Grabisch and Rusinowska, 2010] allow only for positive influence, we adopt a more general notion of influence, which changes the opinion of an agent but not necessarily making it the same as the opinion of the influencing agents. Thus, being influenced just means that an agent modifies his opinion w.r.t. his current inclination. For example an agent could say that "if Bob likes white wine, I would like to take white wine as well", or "if Alice doesn't like pasta, I would like to take pasta".

Moreover, we allow for conditional influence that holds only in a specific context, where the context is the assignment of some variables. For example, an agent could say "if we decide to drink wine, I will follow Bob's preferences, otherwise I will follow my inclination".

Besides this form of influence over the same feature, we also want to allow influence to come from the preferences of other agents over different features. For example, assume a set of friends needs to decide whether to go out together today or tomorrow, and if to have dinner or lunch. Then an agent could say "if Bob prefers to go out tomorrow, I prefer to go for dinner".

In [Grabisch and Rusinowska, 2010] an influence function is a set of statements, or equivalently a matrix or a graph, saying how agents are influenced by each other. We will model each influence function via one or more conditional influence statements.

Definition 2 *A conditional influence statement (ci-statement) on variable X has the form*

$$X_1 = v_1, \ldots, X_k = v_k :: o(X)$$

where $o(X)$ is an ordering over the values of variable X. Variables $X_1, \ldots X_k$ are the influencing variables and variable X is the influenced variable.

A ci-statement $X_1 = v_1, \ldots, X_k = v_k :: o(X)$ models the influence on variable X of an assignment to the set of influencing variables $X_1, \ldots, X_k$. A ci-table is a collection of ci-statements with the same influencing and influenced variables, and containing at most one ci-statement for each assignment of the influencing variables.

As in CP-nets dependencies are graphically denoted by hyperarcs, we also use hyperarcs to graphically denote ci-tables. Such hyperarcs go from the influencing variables to the influenced variable. To distinguish them from the dependencies, we call them ci-arcs.

Definition 3 *An i-profile is a triple (P, O, S), where*

- *P is a profile,*
- *O is an ordering over the m features of the profile, and*
- *S is a set of ci-tables.*

Moreover:

- *The ordering O of the features must be such that $Dep(P)$ has only arcs from earlier variables to later variables. This ordering partitions the set of variables into m levels. Variables in the same level correspond to the same feature.*
- *The ci-tables of an i-profile must be such that each variable can be influenced by variables in her level or in earlier levels, but not in the same ci-statement.*

Notice that, because of the restriction we impose on ci-tables, ci-arcs in an i-profile can create cycles only within variables of the same level.

Example 1 *Consider the i-profile of Figure 1. There are three agents and thus we have three CP-nets. In this example the three CP-nets have the same dependency structure (thus they are obviously compatible). There are two binary features: X and Y, with values, respectively, x and $\bar{x}$, and y*

and $\bar{y}$. The ordering O is $X \succ Y$. Thus the i-profile has six variables denoted by X_1, X_2, X_3, Y_1, Y_2, and Y_3. Each variable X_i (resp., Y_i), with $i \in \{1, 2, 3\}$, has two values denoted by x_i and $\bar{x}_i$ (resp., y_i and $\bar{y}_i$). Notice that values x_i for the variables X_i correspond to value x for X, and similarly for Y. The variables X_i belong to the first level while the variables Y_i belong to the second level. Cp-dependencies are denoted by solid-line arrows and ci-statements are denoted by dotted-line arrows. As it can be seen, agent 3 is influenced (positively) on feature X by agent 2.

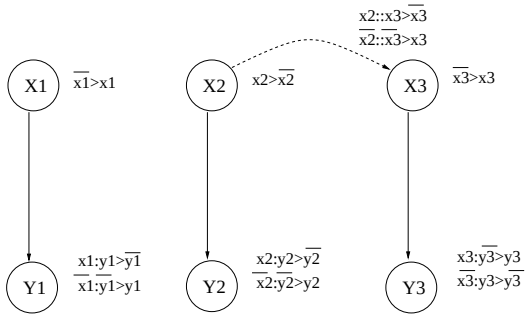


Figure 1: Example of an i-profile.

3.2 Modelling influence functions

Consider the **Conf3** influence function. There is a binary issue to be decided upon, and four people that express their opinions: a king, a man, a woman, and a child. The man follows the king, the woman and child follow the man, and the king is influenced by others only if he has a positive inclination, in which case he will follow such an inclination only if at at least one the other people agrees with him. As shown in [Grabisch and Rusinowska, 2010], this influence function converges to one of two stable states, which both represent consensus, depending on the initial state.

To model this function, we may use a single binary feature X and 4 binary variables X_k, X_m, X_w, and X_c. Each variable X_i, with $i \in \{k, m, w, c\}$, has two values denoted by x_i and $\bar{x}_i$.

The ci-tables representing the influences are:

$King$	Man
$\bar{x}_k - - - :: \bar{x}_k \succ x_k$	$x_k :: x_m \succ \bar{x}_m$
$x_k\bar{x}_m\bar{x}_w\bar{x}_c :: \bar{x}_k \succ x_k$	$\bar{x}_k :: \bar{x}_m \succ x_m$
$x_k x_m - - :: x_k \succ \bar{x}_k$	
$x_k - x_w - :: x_k \succ \bar{x}_k$	
$x_k - - x_c :: x_k \succ \bar{x}_k$	

$Woman$	$Child$
$x_m :: x_w \succ \bar{x}_w$	$x_m :: x_c \succ \bar{x}_c$
$\bar{x}_m :: \bar{x}_w \succ x_w$	$\bar{x}_m :: \bar{x}_c \succ x_c$

A general mapping from any influence function to a set of ci-statements can easily be defined. In general, this mapping will produce between 1 and $n \times 2^n$ ci-statements if we have n agents. In the above example we have exploited the fact that the influence function has a compact formulation in terms of as many influence statements as the number of people involved, and thus we have obtained a much smaller number of ci-statements.

Given an influence function f, we will call $ci(f)$ the ci-statements modelling f.

3.3 Ci- or cp-statements?

It is interesting to notice that ci-statements can be interpreted as cp-statements. In fact, if we see the statements $ci(f)$ as cp-statements, their optimal outcomes coincide with the stable states of the influence function f.

As it is known [Brafman and Dimopoulos, 2004], the optimal outcomes of a set of cp-statements are the solutions of a set of constraints, where each constraint correspond to one of the cp-statements. Following this approach, the constraints corresponding to the statements above are:

- for the king:
  $\bar{x}_k - - - \Rightarrow \bar{x}_k$
  $x_k\bar{x}_m\bar{x}_w\bar{x}_c \Rightarrow \bar{x}_k$
  $x_k x_m - - \Rightarrow x_k$
  $x_k - x_w - \Rightarrow x_k$
  $x_k - - x_c \Rightarrow x_k$
- for the man:
  $x_k \Rightarrow x_m$
  $\bar{x}_k \Rightarrow \bar{x}_m$
- for the woman:
  $x_m \Rightarrow x_w$
  $\bar{x}_m \Rightarrow \bar{x}_w$
- for the child:
  $x_m \Rightarrow x_c$
  $\bar{x}_m \Rightarrow \bar{x}_c$

The only two solutions of this set of constraints are: (x_k, x_m, x_w, x_c) and $(\bar{x_k}, \bar{x_m}, \bar{x_w}, \bar{x_c})$, which are exactly the two stable states of the **Conf3** influence function.

Theorem 1 *Given an influence function f, consider the cp-statements corresponding the ci-statements $ci(f)$. Then the optimal outcomes of $ci(f)$ coincide with the stable states of f.*

In other words, influences and cp-dependencies are not different in their semantics. This is very useful, since it allows for a very simple integration of ci- and cp-statements in the same profile. However, we need to give them a different syntax since we must distinguish between the initial inclination of the agents, given by the cp-statements, and the influences, given by the ci-statements. In fact, influences modify the initial inclination by overriding the preferences, but the opposite does not hold. So it would be a mistake to just treat the ci-statements as additional cp-statements in the profile.

4 Aggregating influenced preferences

We will now propose a way to aggregate the preferences contained in an i-profile, while taking into account the influence functions. The main idea is to use a sequential approach where at each step we consider one of features, in the ordering stated by the i-profile. The method we propose includes three main phases: influence iteration within one level, propagation

from one level to the next one, and preference aggregation. At the end, a winner candidate will be selected, that is, a value for each feature.

In the following subsections, we will describe each of these phases and how they can be combined.

4.1 Influence iteration

For each feature, we consider the influences among different variables modelling this feature and thus belonging to the same level. What we need to do is to find, if it exists, the stable state of such influences corresponding to the initial inclination of the agents. Such inclination is given by the cp-tables of these variables in the profile.

Consider the hypergraph corresponding to the ci-statements over variables representing the same feature. We consider this hypergraph to be cyclic if there are cycles of length at least 2. In fact, a cycle of length 1 models the fact that a variable is influenced by other variables and also by its current inclination.

Notice that, when we are at the first level, the variables are all independent in terms of cp-dependencies, so each agent has an inclination over the values of his variable which does not depend on any other variable.

To find stability or to find out that there is no stable state, we employ an iterative algorithm (see Algorithm 1 below). This algorithm starts with the assignment s of all variables given by their initial inclination, which can be seen in their cp-statements, and moves to another assignment s' by setting the value of each variable to its most preferred value given the values in s of its influencing variables (this is achieved by function ci-flip in Algorithm 1). It then iterates this step until either it reaches a fixpoint or it sees an assignment twice. In the first case, the fixpoint gives us a stable state and the variables are fixed to such values. In the second case, it stops and reports a non-convergent influence for the variables of the considered level.

Algorithm 1: Influence iteration algorithm

```
s = (s_1, ..., s_n) // the initial inclination
s' = s
repeat
    s = s'
    for i=1 to n do
        s'_i = ci-flip(s, i)
until s = s' or s' already seen ;
if s = s' then
    return s
else
    return "No convergence"
```

Notice that, if the ci-statements do not generate cycles, stability is always reached, since the structure is assimilable to an acyclic CP-net, which always has exactly one optimal outcome, thus by Theorem 1 the influence statements have exactly one stable state corresponding to the initial inclination.

4.2 Propagation

Once the variables of a certain level have been fixed to some values, by the influence iteration procedure outlined above, we can propagate to the next level this information by considering the ci- and cp-statements that go from the current level to the next one. Propagation through a ci- or cp-table is achieved by eliminating the conditional statements that refer to conditions not satisfied by the chosen assignment of the influencing or parent variables. The resulting table has exactly one value ordering, giving us the inclination of that variable.

Since influence overrides preference, we first look at the ci-tables and set the inclination of the influenced variables according to such tables. For the variables whose inclination has not been determined after this step, their inclination will be determined by their cp-tables.

After this, we are ready to handle the next level as we did for the first one, since all of its variables are now subject only to influence functions.

4.3 Preference aggregation

In the previous section we have described how to reach stability within one level and how to propagate the decision taken at one level to the next one. It remains to decide when to perform preference aggregation in order to obtain a winner from the profile.

If the influence statements within each level model an influence function which always converges to a consensus state, as it is the case for the **Gur** or the **Conf3** functions, then aggregation is redundant, since all variables at the same level have the same value. Thus the most preferred outcome is the same for all agents, and this will be declared the winner (with any unanimous voting rule).

However, at each level we obtain a possibly different value for the variables modelling the same feature. Now we can either aggregate at each level, and then propagate the result to the next level, or we can aggregate only at the end of the procedure, when each agent will have a most preferred candidate.

If we decide to aggregate at each level, we will choose by majority (since variables are binary) which value to give to all variables of the considered level. Then we propagate such a choice to the next level and start again with an influence iteration. We call LA this method (for *Level Aggregation*).

Otherwise, we can leave the variable values in each level as they are after the influence iteration and proceed with the interleaving of propagation and convergence, until all levels have been handled. At this point, we have a most preferred candidate for each agent, and we can obtain a winning candidate by any voting rule that needs the top choices, such as plurality. We call FA this method (for *Final Aggregation*).

The two approaches yield different results as shown by the following example.

Example 2 *Let us consider the i-profile of Figure 1. After the influence iteration step at level 1 (that is, on feature X), the preference of agent 3 is $x_3 \succ \bar{x}_3$, while the preferences of the other agents are unchanged.*

Assume to adopt method LA. Then we now aggregate the votes over X by majority. This results in $X = x$ winning and

thus the variables of the first level are set to the following values: $X_1 = x_1$, $X_2 = x_2$, and $X_3 = x_3$. We then propagate such assignments to the next level and we get the following assignment for the variables corresponding to the Y feature: $Y_1 = y_1$, $Y_2 = y_2$, and $Y_3 = \bar{y}_3$. We now aggregate the votes over Y by majority, and the winning assignment is $Y = y$. Thus the overall winner of the procedure is $\langle X = x, Y = y\rangle$.

Instead, if we follow the FA procedure, the assignments for X that are propagated are those after the influence iteration, that is, $X_1 = \bar{x}_1$, $X_2 = x_2$, and $X_3 = x_3$. This gives, through propagation, the following values for the variables corresponding to Y: $Y_1 = \bar{y}_1$, $Y_2 = y_2$, and $Y_3 = \bar{y}_3$. Thus we have the following three top candidates for the three agents: $C1 = (X = \bar{x}, Y = \bar{y})$, $C2 = (X = x, Y = y)$, and $C3 = (X = x, Y = \bar{y})$. Now we aggregate, for example by using plurality, with a tie-breaking rule where precedence is given by a lexicographical ordering where $\bar{x} \succ x$ and $\bar{y} \succ y$. According to this rule, the winner is $(X = \bar{x}, Y = \bar{y})$.

Notice that the choice of the ordering does not matter, since, if we consider an i-profile (P, O, S), any other i-profile (P, O', S) will produce the same final result. In fact, different orderings of an i-profile with the same profile and the same ci-statements will order differently variables that are independent both in terms of preferences and influence functions.

However, as seen in the example above, in general the two procedures LA and FA return different winners. Moreover, some agents may be better off with one of the two procedures, while others may be better off with the other one. This is the case of agent 1, that gets its top candidate to win with FA, while it would get a worse candidate with LA. The opposite situation holds for agent 2.

5 Conclusions and future work

In this paper we have assumed that agents express their preferences via CP-nets. We also plan to consider settings where other formalisms for compact preference representation are used, such as soft constraints.

We plan to study the normative properties of procedures LA and FA, as well as to asses their behavior via experimental tests.

In [Grabisch and Rusinowska, 2010] there are also influence functions where influence is followed with a certain probability, otherwise the agent follows its inclination. We plan to study how to generalize our framework to allow for such influence functions.

In [M. Grabisch, 2003] influence is over the top choice among a set of possible actions, not just two. We plan to formalize the extension of our approach to this case. We also plan to allow for influences over the ordering of the actions, rather than just over the top element of such an ordering.

References

[Boutilier *et al.*, 2004] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, 21:135–191, 2004.

[Brafman and Dimopoulos, 2004] R.I. Brafman and Y. Dimopoulos. Extended semantics and optimization algorithms for cp-networks. *Computational Intelligence*, 20(2):218–245, 2004.

[DeGroot, 1974] M.H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69:118–121, 1974.

[Faliszewski *et al.*, 2009] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. How hard is bribery in elections? *J. Artif. Intell. Res. (JAIR)*, 35:485–532, 2009.

[Grabisch and Rusinowska, 2010] Michel Grabisch and Agnieszka Rusinowska. Iterating influence between players in a social network. Documents de travail du centre d'economie de la sorbonne, Universit Panthon-Sorbonne (Paris 1), Centre d'Economie de la Sorbonne, 2010.

[Krause, 2000] U. Krause. A discrete nonlinear and nonautonomous model of consensus formation. *Communications in Difference Equations*, 2000.

[Lang and Xia, 2009] Jerome Lang and Lirong Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical social sciences*, 57:304–324, 2009.

[M. Grabisch, 2003] A. Rusinowska M. Grabisch. A model of influence wuth an ordered set of possible actions. *Theory and Decisions*, 69(4):635–656, 2003.

[P. DeMarzo, 2003] D. Vayanos P. DeMarzo. Persuasion bias, social influence, and unidimensional opinions. *Quarterly Journal of Economics*, 118:909–968, 2003.

[Purrington and Durfee, 2007] K. Purrington and E. H. Durfee. Making social choices from individuals' cp-nets. In *AAMAS*, page 179. IFAAMAS, 2007.

[Rossi *et al.*, 2004] F. Rossi, K.B. Venable, and T. Walsh. mcp nets: Representing and reasoning with preferences of multiple agents. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI 2004)*, pages 729–734, 2004.

[Slinko and White, 2008] Arkadii Slinko and Shaun White. Is it ever safe to vote strategically? Department of mathematics - research reports-563, 2008.

[Xia *et al.*, 2008] L. Xia, V. Conitzer, and J. Lang. Voting on multiattribute domains with cyclic preferential dependencies. In *AAAI*, pages 202–207. AAAI Press, 2008.

Manipulation of Nanson's and Baldwin's Rules

Nina Narodytska
NICTA and UNSW
Sydney, Australia
ninan@cse.unsw.edu.au

Toby Walsh
NICTA and UNSW
Sydney, Australia
toby.walsh@nicta.com.au

Lirong Xia
Department of Computer Science
Duke University
Durham, NC 27708, USA
lxia@cs.duke.edu

Abstract

Nanson's and Baldwin's voting rules select a winner by successively eliminating candidates with low Borda scores. We show that these rules have a number of desirable computational properties. In particular, with unweighted votes, it is NP-hard to manipulate either rule with one manipulator, whilst with weighted votes, it is NP-hard to manipulate either rule with a small number of candidates and a coalition of manipulators. As only a couple of other voting rules are known to be NP-hard to manipulate with a single manipulator, Nanson's and Baldwin's rules appear to be particularly resistant to manipulation from a theoretical perspective. We also propose a number of approximation methods for manipulating these two rules. Experiments demonstrate that both rules are often difficult to manipulate in practice. These results suggest that elimination style voting rules deserve further study.

1 Introduction

Computational social choice studies computational aspects of voting. For example, how does a coalition of agents compute a manipulation? Can we compile these votes into a more compact form? How do we decide if we have elicited enough votes from the agents to be able to declare the result? Whilst there has been a very active research community studying these sort of questions for well known voting rules like plurality and Borda, there are other less well known rules that might deserve attention. In particular, we put forward two historical voting rules due to Nanson and Baldwin which are related to Borda voting.

There are several reasons to consider these two rules. Firstly, they have features that might appeal to the two opposing camps that support Borda and Condorcet. In particular, both rules are Condorcet consistent as they elect the candidate who beats all others in pairwise elections. Secondly, both rules are elimination style procedures where candidates are successively removed. Other elimination procedures like STV and plurality with runoff are computationally hard to manipulate (in the case of STV, with or without weights on the votes, whilst in the case of plurality with runoff, only in the case of weighted votes). We might therefore expect Nanson's and Baldwin's rules to be computationally hard to manipulate. Thirdly, statistical analysis suggests that, whilst the Borda rule is vulnerable to manipulation [7], Nanson's rule is particularly resistant [14]. We might expect Baldwin to be similarly resistant. Finally, the two rules have been used in real elections in the Universitiy of Melbourne (between 1926 and 1982), the University of Adelaide (since 1968), and the State of Michigan (in the 1920s). It is perhaps therefore somewhat surprising that neither rule has received much attention till now in the computational social choice literature.

2 Preliminaries

Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be the set of *candidates* (or *alternatives*). A linear order on $\mathcal{C}$ is a transitive, antisymmetric, and total relation on $\mathcal{C}$. The set of all linear orders on $\mathcal{C}$ is denoted by $L(\mathcal{C})$. An n-voter profile P on $\mathcal{C}$ consists of n linear orders on $\mathcal{C}$. That is, $P = (V_1, \ldots, V_n)$, where for every $j \leq n$, $V_j \in L(\mathcal{C})$. The set of all n-profiles is denoted by $\mathscr{F}_n$. We let m denote the number of candidates. A (deterministic) *voting rule* r is a function that maps any profile on $\mathcal{C}$ to a unique winning candidate, that is, $r : \mathscr{F}_1 \cup \mathscr{F}_2 \cup \ldots \to \mathcal{C}$. In this paper, if not mentioned otherwise, ties are broken in the fixed order $c_1 \succ c_2 \succ \cdots \succ c_m$.

(Positional) scoring rules are commonly used voting rules. Each positional scoring rule is identified by a *scoring vector* $\vec{s}_m = (\vec{s}_m(1), \ldots, \vec{s}_m(m))$ of m integers, for any vote $V \in L(\mathcal{C})$ and any candidate $c \in \mathcal{C}$, let $\vec{s}_m(c, V) = \vec{s}_m(j)$, where j is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $\vec{s}_m(c, P) = \sum_{j=1}^{n} \vec{s}_m(c, V_j)$. The rule selects $c \in \mathcal{C}$ such that the total score $\vec{s}_m(c, P)$ is maximized. We assume scores are integers and decreasing. *Borda* is the positional scoring rule that corresponds to the scoring vector $(m-1, m-2, \ldots, 0)$. We write $s(a, P)$ for the Borda score given to candidate a from the profile of votes P, and $s(a)$ where P is obvious from the context. When voters are weighted (that is, each voter is associated with a positive real number as the weight), a positional scoring rule selects the candidate that maximizes the weighted total score.

The *unweighted (coalitional) manipulation* problem is defined as follows. An instance is a tuple (r, P^{NM}, c, M), where r is a voting rule, P^{NM} is the non-manipulators' profile, c is the candidate preferred by the manipulators, and M

is the set of manipulators. We are asked whether there exists a profile P^M for the manipulators such that $r(P^{NM} \cup P^M) = c$. The *weighted (coalitional) manipulation* is defined similarly, where the weights of the voters (both non-manipulators and manipulators) are also given as inputs. As is common in the literature, we break ties in favour of the coalition of the manipulators where appropriate.

3 Nanson's and Baldwin's Rules

The Borda rule has several good properties. For instance, it is monotonic as increasing the score for a candidate only helps them win. Also it never elects the Condorcet loser (a candidate that loses to all others in a majority of head to head elections). However, it may not elect the Condorcet winner (a candidate that beats all others in a majority of head to head elections). Nanson's and Baldwin's rules, by comparison, always elect the Condorcet winner when it exists.

Nanson's and Baldwin's rules are derived from the Borda rule. Nanson's rule eliminates all those candidates with less than the average Borda score [16]. The rule is then repeated with the reduced set of candidates until there is a single candidate left. A closely related voting rule proposed by Baldwin successively eliminates the candidate with the lowest Borda score[1] until one candidate remains [2]. The two rules are closely related, and indeed are sometimes confused. One of the most appealing properties of Nanson's and Baldwin's rules is that they are Condorcet consistent, i.e. they elect the Condorcet winner. This follows from the fact that the Borda score of the Condorcet winner is never below the average Borda score. Both rules possess several other desirable properties including the majority criterion and the Condorcet loser criterion. There are also properties which distinguish them apart. For instance, Nanson's rule satisfies reversal symmetry (i.e. if there is a unique winner and voters reverse their vote then the winner changes) but Baldwin's rule does not.

4 Unweighted Manipulation

We start by considering the computational complexity of manipulating both these rules with unweighted votes. We prove that the coalitional manipulation problem is NP-complete for both rules even with a single manipulator. Computational intractability with a single manipulator is known only for a small number of other voting rules including the second order Copeland rule [4], STV [3] and ranked pairs [18]. In contrast, when there are two or more manipulators, unweighted coalitional manipulation is hard for some other common voting rules [12; 13; 19; 11; 5]. Our results therefore significantly increase the size of the set of voting rules used in practice that are known to be NP-hard to manipulate with a single manipulator. This also contrasts to Borda where computing a manipulation with a single manipulator is polynomial [4]. Adding elimination rounds to Borda to get Nanson's or Baldwin's rules increases the computational complexity of computing a manipulation with one manipulator from polynomial to NP-hard.

[1] If multiple candidates have the lowest score, then we use a tie-breaking mechanism to eliminate one of them.

Our results are proved by reductions from the EXACT 3-COVER (X3C) problem. An X3C instance contains two sets: $\mathcal{V} = \{v_1, \ldots, v_q\}$ and $\mathcal{S} = \{S_1, \ldots, S_t\}$, where $t \geq 2$ and for all $j \leq t$, $|S_j| = 3$ and $S_j \subseteq \mathcal{V}$. We are asked whether there exists a subset $\mathcal{S}'$ of $\mathcal{S}$ such that each element in $\mathcal{V}$ is in exactly one of the 3-sets in $\mathcal{S}'$.

Theorem 1. *With unweighted votes, the coalitional manipulation problem under Baldwin's rule is NP-complete even when there is only one manipulator.*

Proof: We sketch a reduction from X3C. Given an X3C instance $\mathcal{V} = \{v_1, \ldots, v_q\}, \mathcal{S} = \{S_1, \ldots, S_t\}$, we let the set of candidates be $\mathcal{C} = \{c, d, b\} \cup \mathcal{V} \cup \mathcal{A}$, where c is the candidate that the manipulator wants to make the winner, $\mathcal{A} = \{a_1, \ldots, a_t\}$, and d and b are additional candidates. Members of $\mathcal{A}$ correspond to the 3-sets in $\mathcal{S}$. Let $m = |\mathcal{C}| = q + t + 3$.

The profile P contains two parts: P_1, which is used to control the changes in the score differences between candidates, after a set of candidates are removed, and P_2, which is used to balance the score differences between the candidates. We define the votes $W_{(u,v)} = \{u \succ v \succ \textit{Others}, \text{rev}(\textit{Others}) \succ u \succ v\}$ where *Others* is a total order in which the candidates in $\mathcal{C} \setminus \{u, v\}$ are in a pre-defined lexicographic order, and rev(*Others*) is the reverse.

We make the following observations on $W_{(c_1,c_2)}$. For any set of candidates $\mathcal{C}' \subseteq \mathcal{C}$ and any pair of candidates $e_1, e_2 \in \mathcal{C} \setminus \mathcal{C}'$,

$$\begin{aligned} & s(e_1, W_{(c_1,c_2)}|_{\mathcal{C}\setminus\mathcal{C}'}) - s(e_2, W_{(c_1,c_2)}|_{\mathcal{C}\setminus\mathcal{C}'}) \\ = & \, s(e_1, W_{(c_1,c_2)}) - s(e_2, W_{(c_1,c_2)}) \\ & + \begin{cases} 1 & \text{if } e_1 = c_2 \text{ and } c_1 \in \mathcal{C}' \\ -1 & \text{if } e_1 = c_1 \text{ and } c_2 \in \mathcal{C}' \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Here $W_{(c_1,c_2)}|_{\mathcal{C}\setminus\mathcal{C}'}$ is the pair of votes obtained from W by removing all candidates in $\mathcal{C}'$. In words, the formula states that after $\mathcal{C}'$ is removed, the score difference between e_1 and e_2 is increased by 1 if and only if $e_1 = c_2$ and c_1 is removed; it is decreased by 1 if and only if $e_1 = c_1$ and c_2 is removed; for any other cases, the score difference does not change. Moreover, for any $e \in \mathcal{C} \setminus \{c_1, c_2\}$, $s(c_1, W_{(c_1,c_2)}) - s(e, W_{(c_1,c_2)}) = 1$ and $s(c_2, W_{(c_1,c_2)}) - s(e, W_{(c_1,c_2)}) = -1$.

We next show how to use $W_{(c_1,c_2)}$ to construct the first part of the profile P_1. Let $m = |\mathcal{C}|$, that is, $m = q + t + 3$. P_1 is composed of the following votes: (1) for each $j \leq t$ and each $v_i \in S_j$, there are $2m$ copies of $W_{(v_i,a_j)}$; (2) for each $i \leq q$, there are m copies of $W_{(b,v_i)}$; (3) there are $m(t+6)$ copies of $W_{(b,c)}$. It is not hard to verify that $s(b, P_1) - s(c, P_1) \geq mq$, and for any $c' \in \mathcal{V} \cup \mathcal{A}$, $s(c', P_1) - s(c, P_1) \geq 2m$. P_2 is composed of the following votes: (1) for each $i \leq q$, there are $s(v_i, P_1) - s(c, P_1) - m$ copies of $W_{(d,v_i)}$; (2) for each $j \leq t$, there are $s(a_j, P_1) - s(c, P_1) - 1$ copies of $W_{(d,a_j)}$; (3) there are $s(b, P_1) - s(c, P_1) - mq$ copies of $W_{(d,b)}$.

Let $P = P_1 \cup P_2$. We make the following observations on the Borda scores of the candidates in P.

- For any $i \leq q$, $s(v_i, P) - s(c, P) = m$;
- for any $j \leq t$, $s(a_j, P) - s(c, P) = 1$;

• $s(b, P) - s(c, P) = mq$.

Suppose the X3C instance has a solution, denoted by (after reordering the sets in $\mathcal{S}$) $S_1, \ldots, S_{q/3}$. Then, we let the manipulator vote for:

$c \succ d \succ a_{q/3+1} \succ \cdots \succ a_t \succ b \succ \mathcal{V} \succ a_1 \succ \cdots \succ a_{q/3}$

In the first $4q/3$ rounds, all candidates in $\mathcal{V}$ and $\{a_1, \ldots, a_{q/3}\}$ drop out. Then b drops out. In the following $t - q/3$ rounds the candidates in $\{a_{q/3+1}, \ldots, a_t\}$ drop out. Finally, d loses to c in their pairwise election, which means that c is the winner.

Suppose the manipulator can cast a vote to make c the winner. We first note that d must be eliminated in the final round since its score is higher than c in all previous rounds. In the round when b is eliminated, the score of b should be no more than the score of c. We note that $s(b, P) - s(c, P) = mq$ and the score difference can only be reduced by the manipulator ranking b below c, and by eliminating $v_1, \ldots, v_q$ before b. However, by ranking b below c, the score difference is reduced by no more than $m - 1$. Therefore, before b drops out, all candidates in $\mathcal{V}$ must have already dropped out. We note that for any $v_i \in \mathcal{V}$, $s(v_i, P) - s(c, P) = m$. Therefore, for each $v_i \in \mathcal{V}$, there exists a_j with $v_i \in S_j$ who is removed before v_i. For any such a_j, none of the candidates in S_j can drop out before a_j (otherwise the score of a_j cannot be less than c before b drops out), and in the next three rounds the candidates in S_j drop out. It follows that the set of candidates in $\mathcal{A}$ that drop out before any candidate in $\mathcal{V}$ corresponds to an exact cover of $\mathcal{V}$. □

Theorem 2. *With unweighted votes, the coalitional manipulation problem under Nanson's rule is NP-complete even when there is only one manipulator.*

The proof uses the same gadget $W_{(u,v)}$ that is used in the proof of Theorem 1. Due to the space constraints, the proof can be found in an online technical report.

Weighted Manipulation

If the number of candidates is bounded, then manipulation is NP-hard to compute when votes are weighted. Baldwin's rule appears more computationally difficult than Nanson's rule. Coleman and Teague [8] prove that Baldwin's requires only 3 candidates to be NP-hard, whilst we prove here that Nanson's rule is polynomial to manipulate with 3 candidates and requires at least 4 candidates to be NP-hard. It follows that computing a manipulation is NP-hard for both rules when votes are unweighted, the number of candidates is small and there is uncertainty about how agents have voted in the form of a probability distribution [9]. Note that the coalition manipulation problem for Borda with weighted votes is NP-hard for 3 or more candidates [9]. Thus, somewhat surprisingly, adding an elimination round to Borda, which gives us Nanson's rule, decreases the computational complexity of computing a manipulation with 3 manipulators from NP-hard to polynomial.

Theorem 3. *With Nanson's rule and weighted votes, the coalition manipulation problem is NP-complete for just 4 candidates.*

Proof: The proof is by a reduction from PARTITION, where we are given a group of integers $\{k_1, \ldots, k_l\}$ with sum $2K$, and we are asked whether there is way to partition the group into two groups, the elements in each of which sum to K. For any PARTITION instance, we construct a coalition manipulation problem with 4 candidates (a, b, c and p) where p is again the candidate that the manipulators wish to win. We suppose the non-manipulators have voted as follows: $2K + 1$ for each of $b \succ p \succ c \succ a$, $a \succ c \succ b \succ p$, $c \succ p \succ b \succ a$ and $a \succ b \succ c \succ p$, $K + 2$ for $p \succ a \succ b \succ c$ and $c \succ b \succ p \succ a$, and 1 each for $a \succ b \succ p \succ c$, $c \succ p \succ a \succ b$, $a \succ c \succ p \succ b$ and $b \succ p \succ a \succ c$. The total scores from non-manipulators are as follows: $s(a) = 14K + 18$, $s(b) = s(c) = 17K + 18$ and $s(p) = 12K + 18$. For each integer k_i, we have a member of the manipulating coalition with weight k_i.

Now, suppose there is a solution to the PARTITION instance. Let the manipulators corresponding to the integers in one half of the partition vote $p \succ a \succ b \succ c$, and let the others vote $p \succ a \succ c \succ b$. All scores are now $18K + 18$ (which is also the average). By the tie-breaking rule, p wins in the first round. Thus the manipulators can make p win if a perfect partition exists.

Conversely, suppose there is a successful manipulation. Clearly, p cannot be eliminated in the first round. To ensure this, all manipulators must put p in first place. Next, we show that if p is not a joint winner of the first round, p cannot win overall. We consider all possible sets of candidates that could be eliminated in the first round. There are 6 cases. In the first case, only a is eliminated in the first round. The scores from non-manipulators in the second round are as follows: $s(b) = s(c) = 12K + 13$, and $s(p) = 6K + 10$. The average score is $10K + 12$. Even with the maximum $4K$ possible score from the manipulators, p is eliminated. This contradicts the assumption that p wins. In the second case, only b is eliminated in the first round. As p and a are not eliminated in the first round, the manipulators have to cast votes that put p in first place and b in second place. With such manipulating votes, the scores in the second round are: $s(a) = 11K + 11$, $s(c) = 12K + 12$ and $s(p) = 13K + 13$. The average score is $12K + 12$. Hence, a is eliminated. In the next round, p is eliminated as $s(p) = 5K + 5$, $s(c) = 7K + 7$ and the average score is $6K + 6$. This contradicts the assumption that p wins. In the third case, only c is eliminated in the first round. This case is symmetric to the second case. In the fourth case, a and b are eliminated in the first round. The case when a and c are eliminated is symmetric. In the second round, the scores from non-manipulators are $s(c) = 7K + 7$ and $s(p) = 3K + 5$. The $2K$ score from the manipulators cannot prevent p being eliminated. This contradicts the assumption that p wins. In the fifth case, b and c are eliminated in the first round. However, in the first round, the score b and c receive from the non-manipulators is $17K + 18$. One of them will get at least K points from manipulators. This will give them greater than the average score of $18K + 8$. Hence, at least one of them is not eliminated. In the sixth and final case, a, b and c are all eliminated in the first round. This case is again impossible by the same argument as the last case.

The only way for p to win is to have a tie with all candidates in the first round. As we observed above, the manipulators

have to put p in first place, and a in second place. In turn, both b and c have to get exactly K points from the manipulators. Hence, there exists a solution to the PARTITION instance. □

Clearly, it is polynomial to compute a manipulation of Baldwin's rule with 2 candidates (since this case degenerates to majority voting). With Nanson's rule, on the other hand, it is polynomial with up to 3 candidates.

Theorem 4. *With Nanson's rule and weighted votes, the coalition manipulation problem is polynomial for up to 3 candidates.*

Proof: Consider an election with 3 candidates (a, b and p) in which the manipulators want p to win. We prove that the optimal strategy is for the manipulators either all to vote $p \succ a \succ b$ or all to vote $p \succ b \succ a$. If p does not win using one of these two votes, then p cannot win. Therefore we simply try out the two votes and compute if p wins in either case.

Suppose the manipulators can make p win. We first note that there is no loss for them to raise p to the first position, while keeping the other parts of their preferences the same. By doing so, the score of p goes up and the scores of a and b go down. The only possible change in the elimination process is that now both a and b drop out in the first round, so that p still wins.

Now, suppose that all manipulators rank p in their top positions. Let P^M denote the manipulators' profile that makes p win. Because Nanson's rule never selects the Condorcet loser, p cannot be beaten by both a and b in pairwise elections. Without loss of generality, suppose p beats a. We argue that if all manipulators vote $p \succ a \succ b$, then p still wins. For the sake of contradiction, suppose all manipulators vote $p \succ a \succ b$ but p does not win. As the manipulators still rank p in their top positions, the score of p in the first round is the same as in P^M. Therefore, p must enter (and lose) the second round. Hence, only a is eliminated in the first round, and in the second round b beats p. However, having the manipulators vote $p \succ a \succ b$ only lowers b's score in the first round, compared to the case where they vote P^M. Hence, when the manipulators vote P^M, b also enters the second round and then beats p, which is a contradiction.

Therefore, if the manipulators can make p win, then they can make p win by all voting $p \succ a \succ b$, or all voting $p \succ b \succ a$. □

5 Approximation Methods

One way to deal with computational intractability is to treat computing a manipulation as an optimization problem where we try to minimize the number of manipulators. We therefore considered five approximation methods. These are either derived from methods used with Borda or are specifically designed for the elimination style of Nanson's and Baldwin's rules.

REVERSE**:** The desired candidate is put first, and the other candidates are reverse ordered by their current Borda score. We repeat this construction until the desired candidate wins. REVERSE was used to manipulate the Borda rule in [20].

LARGESTFIT**:** This method was proposed for the Borda rule [10]. Unlike REVERSE which constructs votes one by one, we construct votes in any order using a bin packing heuristic which puts the next largest Borda score into the "best" available vote. We start with a target number of manipulators. Simple counting arguments will lower bound this number, and we can increase it until we have a successful manipulation. We construct votes for the manipulators in which the desired candidate is in first place. We take the other Borda scores of the manipulators in decreasing order, and assign them to the candidate with the lowest current Borda score who has been assigned less than the required number of scores. A perfect matching algorithm then converts the sets of Borda scores for the candidates into a set of manipulating votes.

AVERAGEFIT**:** This method was also proposed for the Borda rule [10]. We again have a target number of manipulators, and construct votes for the manipulators in which the desired candidate is in first place. We take the other Borda scores of the manipulators in decreasing order, and assign them to the candidate with the current lowest *average* Borda score who has less than the required number of scores. The intuition is that if every score was of average size, we would have a perfect fit. If more than one candidate has the same lowest average Borda score and can accommodate the next score, we tie-break on the candidate with the fewest scores. Examples of LARGESTFIT and AVERAGEFIT can be found in [10].

ELIMINATE**:** We repeatedly construct votes in which the desired candidate is put in first place, and the other candidates in the reverse of the current elimination order. For instance, the first candidate eliminated is put in last place. For Nanson's rule, we order candidates eliminated in the same round by their Borda score in that round.

REVELIMINATE**:** We repeatedly construct votes in which the desired candidate is put in first place, and the other candidates in the current elimination order. For instance, the first candidate eliminated is put in second place. For Nanson's rule, we order candidates eliminated in the same round by the inverse of their Borda score in that round.

The intuition behind ELIMINATE is to move the desired candidate up the elimination order whilst keeping the rest of the order unchanged. With REVELIMINATE, the intuition is to move the desired candidate up the elimination order, and to assign the largest Borda scores to the least dangerous candidates. It is easy to show that all methods will eventually compute a manipulation of Nanson's or Baldwin's rule in which the desired candidate wins.

With Borda voting, good bounds are known on the quality of approximation that is achievable. In particular, [20] proved that REVERSE never requires more than one extra manipulator than optimal. Baldwin's and Nanson's rules appear more difficult to approximate within such bounds. We can give examples where all five methods compute a manipulation that use several more manipulators than is optimal. Indeed, even

Table 1: Percentage of random uniform elections with 5 candidates where the heuristic finds the optimal manipulation.

Rules	REV	LAFIT	AVFIT	ELIM	REVELIM
Baldwin	74.4%	74.4%	**75.8%**	62.2%	75.2%
Nanson	74.6%	76.0%	**78.0%**	65.4%	66.9%
Borda	95.7%	98.8%	**99.8%**	95.7%	10.7%

Table 2: Percentage of urn elections with 5 candidates where the heuristic finds the optimal manipulation.

Rules	REV	LAFIT	AVFIT	ELIM	REVELIM
Baldwin	75.1%	75.4%	**77.3%**	68.9%	83.4%
Nanson	78.1%	79.0%	**79.8%**	72.2%	79.4%
Borda	96.1%	92.7%	**99.9%**	96.1%	4.4%

with a fixed number of candidates, REVERSE can require an unbounded number of extra manipulators.

Theorem 5. *With Baldwin's rule, there exists an election with 7 candidates and $42n$ votes where* REVERSE *computes a manipulation with at least n more votes than is optimal.*

Proof: (Sketch) Consider an election over a, b, c, d, e, f and p where p is the candidate that the manipulators wish to win. We define $R(u, v)$ as the pair of votes: $u \succ v \succ \mathit{Others} \succ p$, rev($\mathit{Others}$) $\succ u \succ v \succ p$ where *Others* is some fixed ordering of the other candidates and rev(*Others*) is its reverse. The non-manipulators cast the following votes: $3n$ copies of $R(a, b)$, $R(b, c)$, $R(c, d)$, $R(d, e)$ and $R(e, f)$. In addition, there are $6n$ copies of the votes: $p \succ a \succ \mathit{Others}$ and rev(Others) $\succ p \succ a$. If $18n$ manipulators vote identically $p \succ a \succ \ldots \succ f$ then p wins. This provides an upper bound on the size of the optimal manipulation. After the non-manipulators have voted, $s(a) = s(f) = 138n$, $s(b) = s(c) = s(d) = s(e) = 141n$ and $s(p) = 42n$. REVERSE will put p in first place. We suppose n is a multiple of 2, but more complex arguments can be given in other cases. After n manipulating votes have been constructed, the scores of candidates a to f are level at $285n/2$ and p is leveled at $48n$. From then on, the manipulators put p in first place and alternate the order of the other candidates. At least $32n$ votes are therefore required for p to move out of last place. □

Asymptotically this result is as bad as we could expect. Any election can be manipulated with $O(n)$ votes by simply reversing all previous votes, and this proof demonstrates that REVERSE may use $O(n)$ more votes than is optimal.

6 Experimental Results

To test the difficulty of computing manipulations in practice and the effectiveness of these approximation methods, we ran some experiments using a similar setup to [17]. We generated either uniform random votes or votes drawn from a Polya Eggenberger urn model. In the urn model, votes are drawn from an urn at random, and are placed back into the urn along with a other votes of the same type. This captures varying degrees of social homogeneity. We set $a = m!$ so that there is a 50% chance that the second vote is the same as the first.

Our first set of experiments used 3000 elections with 5 candidates and 5 non-manipulating voters. This is small enough to find the optimal number of manipulators using brute force search, and thus to determine how often a heuristic computes the optimal solution. We threw out the 20% or so of problems generated in which the chosen candidate has already won before the manipulators vote. Results are given in Tables 1–2. Heuristics that are very effective at finding an optimal manipulation with the Borda rule do not perform as well with Baldwin's and Nanson's rules. For example, AVERAGEFIT almost always finds an optimal manipulation of the Borda rule but can only find an optimal solution about 3/4 of the time with Baldwin's or Nanson's rules.

Table 3: Uniform elections using Baldwin rule. This (and subsequent) tables give the average number of manipulators.

n	Rev	LaFit	AvgFit	Elim	RevElim
4	2.25	2.25	2.25	2.44	**2.21**
8	**2.99**	3.07	3.01	3.35	3.06
16	**4.31**	4.41	4.40	4.79	4.67
32	**5.93**	6.03	6.14	6.61	6.84
64	**8.56**	8.65	8.84	9.54	11.02
128	**12.13**	12.24	12.41	13.37	16.06

Table 4: Uniform elections using Nanson rule.

n	Rev	LaFit	AvgFit	Elim	RevElim
4	**2.15**	2.17	2.15	2.25	2.28
8	2.91	2.96	**2.84**	3.05	3.21
16	4.13	4.27	**4.05**	4.44	4.99
32	**5.80**	5.88	5.81	6.18	7.46
64	**8.51**	8.58	8.82	8.99	12.04
128	**12.07**	12.09	13.00	12.60	17.90

Table 5: Urn elections using Baldwin rule.

n	Rev	LaFit	AvgFit	Elim	RevElim
4	3.26	3.23	3.24	3.35	**3.14**
8	5.95	5.96	5.99	6.37	**5.82**
16	11.64	11.66	11.87	12.74	**11.52**
32	**21.70**	21.78	22.35	24.67	22.41
64	**43.09**	43.37	44.24	49.07	45.70
128	82.19	**81.82**	83.62	95.37	91.80

Table 6: Urn elections using Nanson rule.

n	Rev	LaFit	AvgFit	Elim	RevElim
4	3.20	**3.19**	3.20	3.28	3.22
8	**5.93**	5.98	5.95	6.13	6.09
16	**11.62**	11.93	11.64	12.16	12.37
32	**22.36**	22.78	22.53	24.00	24.39
64	**44.56**	45.50	44.77	48.81	49.69
128	87.18	87.55	**86.76**	97.02	99.43

Our second set of experiments used larger problems. This amplifies the differences between the different approximation methods (but means we are unable to compute the optimal manipulation using brute force search). Problems have between 2^2 and 2^7 candidates, and the same number of votes as candidates. We tested 6000 instances, 1000 at each problem size. Tables 3–6 show the results for the average number of manipulators. The results show that overall REVERSE works slightly better than LARGESTFIT and AVERAGEFIT, which themselves outperform the other two methods especially for problems with large number of candidates. We observe a similar picture with Nanson's rule. This contrasts with the Borda rule where LARGESTFIT and AVERAGEFIT do much better

than REVERSE [10]. In most cases AVERAGEFIT is less effective than LARGESTFIT except urn elections with Nanson's rule.

These experimental results suggest that Baldwin's and Nanson's rules are harder to manipulate in practice than Borda. Approximation methods that work well on the Borda rule are significantly less effective on these rules. Overall, REVERSE, LARGESTFIT and AVERAGEFIT appear to offer the best performance, though no heuristic dominates.

7 Other Related Work

Bag, Sabourian and Winter [1] prove that a class of voting rules which use repeated ballots and eliminate one candidate in each round are Condorcet consistent. They illustrate this class with the *weakest link* rule in which the candidate with the fewest ballots in each round is eliminated. Geller [15] has proposed a variant of single transferable vote where first place votes, candidates are successively eliminated based on their *original* Borda score. Unlike Nanson's and Baldwin's rules, this method does not recalculate the Borda score based on the new reduced set of candidates. For any Condorcet consistent rule (and thus for Nanson's and Baldwin's rule), Brandt et al. [6] showed that many types of control and manipulation are polynomial to compute when votes are single peaked.

8 Conclusions

With unweighted votes, we have proven that Nanson's and Baldwin's rules are NP-hard to manipulate with one manipulator. This increases by two thirds the number of rules known to be NP-hard to manipulate with just a single manipulator. With weighted votes, on the other hand, we have proven that Nanson's rule is NP-hard to manipulate with just a small number of candidates and a coalition of manipulators. We have also proposed a number of approximation methods for manipulating Nanson's and Baldwin's rules. Our experiments suggest that both rules are difficult to manipulate in practice. There are many other interesting open questions coming from these results. For example, are there other elimination style voting rules which are computationally difficult to manipulate? As a second example, with Nanson's and Baldwin's rule what is the computational complexity of other types of control like the addition/deletion of candidates, and the addition/deletion of voters? As a third example, we could add elimination rounds to other scoring rules. Do such rules have interesting computational properties?

Acknowledgments

Nina Narodytska is supported by the Asian Office of Aerospace Research and Development through grant AOARD-104123. Toby Walsh is funded by the Australian Department of Broadband, Communications and the Digital Economy and the ARC. Lirong Xia acknowledges a James B. Duke Fellowship and Vincent Conitzer's NSF CAREER 0953756 and IIS-0812113, and an Alfred P. Sloan fellowship for support. We thank all AAAI-11 and WSCAI reviewers for their helpful comments and suggestions.

References

[1] Bag, P.; Sabourian, H.; and Winter, E. 2009. Multi-stage voting, sequential elimination and Condorcet consistency. *Journal of Economic Theory* 144(3):1278 – 1299.

[2] Baldwin, J. 1926. The technique of the Nanson preferential majority system of election. *Trans. and Proc. of the Royal Society of Victoria* 39:42–52.

[3] Bartholdi, III, J., and Orlin, J. 1991. Single transferable vote resists strategic voting. *Social Choice and Welfare* 8(4):341–354.

[4] Bartholdi, J.J.; Tovey, C.A.; and Trick, M.A. 1989. The Computational Difficulty of Manipulating an Election. *Social Choice and Welfare* 6(3): 227–241.

[5] Betzler, N.; Niedermeier, R.; and Woeginger, G. 2011. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *IJCAI-11*.

[6] Brandt, F.; Brill, M.; Hemaspaandra, E.; and Hemaspaandra, L. 2010. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *AAAI-10*, 715–722.

[7] Chamberlin, J. 1985. An investigation into the relative manipulability of four voting systems. *Behavioural Science* 30:195–203.

[8] Coleman, T., and Teague, V. 2007. On the complexity of manipulating elections. In *CATS-07*, 25–33.

[9] Conitzer, V.; Sandholm, T.; and Lang, J. 2007. When are elections with few candidates hard to manipulate? *JACM* 54(3):1–33.

[10] Davies, J.; Katsirelos, G.; Narodytska, N.; and Walsh, T. 2010. An empirical study of Borda manipulation. In *COMSOC-10*.

[11] Davies, J.; Katsirelos, G.; Narodytska, N.; and Walsh, T. 2011. Complexity of and Algorithms for Borda Manipulation. In *AAAI-11*.

[12] Faliszewski, P.; Hemaspaandra, E.; and Schnoor, H. 2008. Copeland voting: ties matter. In *AAMAS-08*, 983–990.

[13] Faliszewski, P.; Hemaspaandra, E.; and Schnoor, H. 2010. Manipulation of Copeland elections. In *AAMAS-10*, 367–374.

[14] Favardin, P., and Lepelley, D. 2006. Some further results on the manipulability of social choice rules. *Social Choice and Welfare* 26:485–509.

[15] Geller, C. 2005. Single transferable vote with Borda elimination: proportional representation, moderation, quasi-chaos and stability. *Electoral Studies* 24(2):265 – 280.

[16] Nanson, E. 1882. Methods of election. *Trans. and Proc. of the Royal Society of Victoria* 19:197 – 240.

[17] Walsh, T. 2010. An empirical study of the manipulability of single transferable voting. In *ECAI-10*, 257–262.

[18] Xia, L.; Zuckerman, M.; Procaccia, A..; Conitzer, V.; and Rosenschein, J. 2009. Complexity of unweighted coalitional manipulation under some common voting rules. In *IJCAI-09*, 348–353.

[19] Xia, L.; Conitzer, V.; and Procaccia, A. D. 2010. A scheduling approach to coalitional manipulation. In *EC-10*, 275–284.

[20] Zuckerman, M.; Procaccia, A.; and Rosenschein, J. 2009. Algorithms for the coalitional manipulation problem. *Artificial Intelligence* 173(2):392–412.

Voting Power, Hierarchical Pivotal Sets, and Random Dictatorships

David M. Pennock
Yahoo! Research New York
111 West 40th Street, 17th floor
New York, NY 10018
pennockd@yahoo-inc.com

Lirong Xia
Department of Computer Science
Duke University
Durham, NC 27708, USA
lxia@cs.duke.edu

Abstract

In many traditional social choice problems, analyzing the voting power of the voters in a given profile is an important part. Usually the voting power of an agent is measured by whether the agent is *pivotal*. In this paper, we introduce two extensions of the set of pivotal agents to measure agents' voting power in a given profile. The first, which is called *hierarchical pivotal sets*, captures the voting power for an agent to make other agents pivotal. The second, which is called *coalitional pivotal sets*, is based on the fact that each agent is given a weight that is computed similarly to the *Shapley-Shubik power index*. We also introduce random dictatorships induced by the two types of pivotal sets to approximate full random dictatorships. We show that the random dictatorships induced by the hierarchical pivotal sets are *strategic-pivot-proof*, that is, no agent can make herself become one of the possible dictators by voting differently.

We then focus on the hierarchical pivotal sets when the hierarchical level goes to infinity. We prove that for any voting rule that satisfies *anonymity* and *unanimity*, and for any given profile, the union of the hierarchical pivotal sets are a sound and complete characterization of the non-redundant agents. We also show that if the voting rule does not satisfy anonymity, then this characterization might not be complete. Finally, we investigate algorithmic aspects of computing the hierarchical pivotal sets.

1 Introduction

Voting has been used in multiagent systems as a popular way to aggregate agents' preferences over a set of alternatives. Recently, a burgeoning field *computational social choice* was formed to study the computational aspects of voting. In computational social choice, one central problem is to investigate the possibility of using computational complexity as a barrier against manipulation. Researchers have been interested in the computational complexity of computing whether a single agent or a coalition of agents have enough power to replace the winner with their favorite alternative by casting votes strategically in collaboration. See [6] and [8] for nice recent surveys.

Looking back in the literature, the study of voting power has been favored in Political Science and Economics for a long time. It has been playing a central role in at least two other main research directions in addition to the study of manipulation. The first direction is the study of rational choice of voters, motivated by the "paradox of not voting", which dates back to Downs' seminal work [5]. The paradox states that when the number of voters is large, the voting power for a single voter to influence the outcome is negligible. Therefore, nobody should bother to vote, which sharply contradicts the much higher turnout in real-life elections. The paradox of not voting has influenced the study of voting in Political Science for more than half a century, and is still popular nowadays. Many research papers have been devoted to explaining the paradox from both theoretical and empirical sides, yet none of them has been successful so far. See [9; 10] for recent surveys.

The second research direction is the study of a class of coalitional games called *weighted voting games*. In a weighted voting game, each voter has a weight, and a coalition of voters is winning if the sum of their weights is higher than a quota (which is usually set to be half of the total weight). It is important to study the power of the voters for many purposes, e.g., for dividing the profit. One of the most important measurements is the *Shapley-Shubik power index* [13], where a voter's power is measured by (informally speaking) her marginal contribution in making coalitions of voters win.

In all the above research directions, a voter's voting power is determined by whether or not she is *pivotal*. That is, in a given profile, a voter is pivotal if and only if she can change the winner by casting a different vote, assuming that the other voters do not change their votes[1]. However, the mere "pivotal or not" measurement is often not discriminative enough. As the paradox of not voting says, the set of pivotal voters is always too small or even empty when the number of voters is large. This argument is supported by some recent work on the probability that a coalition of voters have power to change the outcome [12; 14].

Our conceptual contributions. In this paper, we introduce two new ways to measure a voter's power in a given profile for

[1] In the study of voting power, we do not consider the voter's incentive to cast a different vote.

a given voting rule. Both ways are extensions of the set of pivotal agents, and are much more discriminative. Therefore, we believe that these extensions provides new angles of the voters' strategic behavior in the three traditional research directions mentioned above. The first extension, which is called *hierarchical pivotal sets*, captures the power for a voter to make other voters pivotal. Given a profile, the level-1 hierarchical set is composed of all pivotal voters; for any $k \geq 2$, the level-k hierarchical set is composed of all voters who can change the level-$(k-1)$ hierarchical set by voting differently. The second extension is called *coalitional pivotal sets*. Such sets are subsets of voters who can change the winner by voting differently in collaboration. Based on the coalitional pivotal sets, we define power indices for the voters similarly to the Shapley-Shubik power index[2].

Our technical contributions. To illustrate the applications of these extensions, we define random dictatorships based on them to approximate the fully random dictatorship (which first chooses a voter uniformly at random, then select the winner to be the top-ranked alternative of the chosen voter). Fully random dictatorship is the only randomized voting rule that satisfies anonymity, Pareto-optimality, and strategy-proofness [11]. We prove that the random dictatorships based on hierarchical pivotal sets are *strategic-pivot-proof*, that is, no agent can make herself one of the possible dictators by voting differently.

Our main technical contribution is the following characterization of the hierarchical pivotal sets. We prove that for any voting rule that satisfies *anonymity* and *unanimity* and any profile, the voters in the hierarchical pivotal sets are not redundant (a voter is redundant if he/she is never pivotal in any profile). And conversely, any non-redundant voter must be in the level-k pivotal set for some $k \leq n+1$, where n is the number of voters. Therefore, in terms of hierarchical pivotal sets, for any anonymous voting rule, in any profile, any voter has some voting power to (directly or indirectly) change the winner. This provides a new perspective towards understanding the paradox of not voting. However, we also show that there exists a voting rule that does not satisfy anonymity, such that for any given profile, not all non-redundant voters are in the union of all hierarchical pivotal sets.

Finally, we investigate algorithmic aspects of computing the hierarchical pivotal sets.

2 Preliminaries

Let $\mathcal{C}$ be a finite set of *alternatives* (or *candidates*). A *vote* V is a linear order over $\mathcal{C}$, i.e., a transitive, antisymmetric, and total relation over $\mathcal{C}$. The set of all linear orders over $\mathcal{C}$ is denoted by $L(\mathcal{C})$. An n-voter profile P over $\mathcal{C}$ is a collection of n linear orders over $\mathcal{C}$, that is, $P = (V_1, \ldots, V_n)$, where for every $j \leq n$, $V_j \in L(\mathcal{C})$. In this paper, we let m denote the number of alternatives and let n denote the number of voters (agents) in a profile. Let $N = \{1, \ldots, n\}$. For any subset $S \subseteq N$, we let P_S denote the sub-profile of P that consists of the votes of the voters in S; let $P_{-S} = P_{N \setminus S}$. When $S = \{i\}$, we write P_{-i} instead of $P_{-\{i\}}$. The set of all n-profiles over $L(\mathcal{C})$ is denoted by $\mathrm{F}_n(\mathcal{C})$. In this paper, a *(voting) rule* r maps any n-profile to a single winning alternative, called the *winner*. Some commonly used voting rules are listed below.

• *Positional scoring rules.* Given a *scoring vector* $\vec{v} = (v_1, \ldots, v_m)$ of m integers, for any vote $V \in L(\mathcal{C})$ and any $c \in \mathcal{C}$, let $s_{\vec{v}}(V, c) = v_i$, where i is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $s_{\vec{v}}(P, c) = \sum_{j=1}^{n} s_{\vec{v}}(V_j, c)$. The rule will select an alternative $c \in \mathcal{C}$ so that $s_{\vec{v}}(P, c)$ is maximized. Some examples of positional scoring rules are *plurality*, for which the scoring vector is $(1, 0, \ldots, 0)$, and *veto*, for which the scoring vector is $(1, \ldots, 1, 0)$. Plurality is also called *majority* when there are only two alternatives.

• *Single transferable vote (STV).* The election has m rounds. In each round, the alternative that gets the minimal plurality score drops out, and is removed from all of the votes. The last-remaining alternative is the winner.

• *Ranked pairs*. This rule first creates an entire ranking of all the alternatives. Let $D_P(c_i, c_j)$ denote the number of votes where $c_i \succ c_j$ minus the number of votes where $c_j \succ c_i$ in the profile P. In each step, we consider a pair of alternatives c_i, c_j that we have not previously considered, which has the highest $D_P(c_i, c_j)$ among the remaining pairs. We then fix the order $c_i \succ c_j$, unless it violates transitivity. We continue until all pairs of alternatives have been considered. The alternative at the top of the ranking wins.

• *Dictatorship*. For every $n \in \mathbb{N}$ there exists a voter $j \leq n$ such that the winner is always the alternative that is ranked in the top position in V_j. Voter j is called a *dictator*.

A voting rule r satisfies *anonymity*, if the winner under r does not depend on the name of the voters. That is, for any permutation M over N and any profile $P = (V_1, \ldots, V_n)$, we have $r(P) = r(M(P)) = r(V_{M(1)}, \ldots, V_{M(n)})$. r satisfies *unanimity*, if for any profile P in which all voters rank the same alternative c in their top positions, $r(P) = c$.

In this paper, we let a *random dictatorship* denote a mapping $D_r : \mathrm{F}_n(\mathcal{C}) \to 2^N$, where r is a "default" voting rule that is used to select the winner in case $D_r(P) = \emptyset$. That is, D_r selects a set of "possible dictators" to be randomized over. D_r naturally induces a mapping that assigns each profile to a probability distribution over $\mathcal{C}$ as follows. For any profile P, if $D_r(P) = \emptyset$, then it selects $r(P)$ with probability 1; if $D_r(P) \neq \emptyset$, then it first selects a voter j from $D_r(P)$ uniformly at random, then let the winner be the top-ranked alternative in V_j. A *fully random dictatorship* is a random dictatorship that always outputs N. A *weighted random dictatorship* D_r^w maps a profile to a probability distribution over N, or $\emptyset$. Similarly to random dictatorships, a weighted random dictatorship naturally induces a mapping that assigns each profile to a probability distribution over $\mathcal{C}$: if $D_r^w(P) = \pi \neq \emptyset$, then it selects a voter j from $D_r^w(P)$ according to the distribution π and let the winner to be the top-ranked alternative in V_j; and if $D_r^w(P) = \emptyset$, then it selects $r(P)$ with probability 1.

3 Pivotal sets and random dictatorships

In this section, we introduce two extensions of pivotal sets and their induced (weighted) random dictatorships, and discuss their relationships.

[2]The concept of coalitional pivotal sets is not new, for example, it is implicitly considered in the coalitional manipulation problems. However, as far as we know, this is the first time it is used to define voting power.

3.1 Hierarchical pivotal sets

Given a voting rule r and a profile P, we define the level-1 pivotal set $\text{PS}_r^1(P) \subseteq N$ to be the set of all pivotal voters. That is, $j \in \text{PS}_r^1(P)$ if and only if there exists a vote V_j' such that $r(P_{-j}, V_j') \neq r(P)$. Let the level-1 random dictatorship $D_r^1(P)$ be a mapping such that $D_r^1(P) = \text{PS}_r^1(P)$.

We argue that D_r^1 prevents voters' strategic behavior to some extent, by showing that any voter j who is not in $D_r^1(P)$ cannot make herself become a member in $D_r^1(P_{-j}, V_j')$ by casting a different vote V_j'. By definition, voter j is not pivotal. Therefore, for any pair of votes V_j' and V_j^*, $r(P_{-j}, V_j') = r(P_{-j}, V_j^*)$, which means that $j \notin D_r^1(P_{-j}, V_j')$. Formally, we have the following definition for random dictatorships.

Definition 1 *A random dictatorship D_r is* strategic-pivot-proof*, if for any profile P, any voter j, and any vote V_j', we have $j \in D_r(P_{-j}, V_j') \Longrightarrow j \in D_r(P)$.*

That is, D_r is strategic-pivot-proof if for any profile, any voter who is not selected by D_r cannot cast a different vote to make himself/herself one of the possible dictators. Of course for a strategic-pivot-proof random dictatorship, the voter might still have power and incentive to cast a different vote to change the set of possible dictators, even though she is not in it anyway. Therefore, it seems that strategic-pivot-proofness is weaker than the usual strategy-proofness. We note that they are actually not comparable. Exploring their relationship is an interesting direction for future research.

The level-1 pivotal set and its induced random dictatorship are not the end of the story. To capture the voting power for a voter to change the level-1 pivotal set, we can define level-2 pivotal sets to be composed of all voters who can change the level-1 pivotal set by voting differently. More generally, for any natural number k, we define the level-k pivotal set $\text{PS}_r^k(P) \subseteq N$ recursively as follows.

Definition 2 *For any voting rule r, any $k \in \mathbb{N}$, and any profile P, we define the level-k pivotal set $PS_r^k(P) \subseteq N$ recursively as follows.*

- *$j \in PS_r^1(P)$ if and only if there exists a vote V_j' such that $r(P) \neq r(P_{-j}, V_j')$.*
- *$j \in PS_r^k(P)$ if and only if there exists a vote V_j' such that $PS_r^{k-1}(P) \neq PS_r^{k-1}(P_{-j}, V_j')$. That is, voter j can change the level-$(k-1)$ pivotal set by voting differently.*

Here k is called the *hierarchical level.* Level-k pivotal sets capture voters' indirect power in the current profile P. The higher the hierarchical level is, the more indirectly the voters in it can influence the outcome for P. We note that the level-k pivotal sets for different profiles can be different.

Let D_r^k denote the random dictatorship such that $D_r^k(P) = \bigcup_{i=1}^k \text{PS}_r^i(P)$. In Section 4 we will show that for any voting rule r that satisfies anonymity and unanimity, D_r^k is an approximation to the fully random dictatorship after all redundant voters are removed. We note that the fully random dictatorship is strategy-proof.

Example 1 *There are two alternatives $\{a, b\}$, 5 voters, and we use the majority rule. Table 1 shows the level-k pivotal sets for all profiles, for $k = 1, 2, 3, 4$. Because the majority rule is anonymous, as we will show later in the paper (Lemma 1), the level-k pivotal set can be represented by a set of votes instead of a set of voters. A pivotal set is denoted by "b" if it is exactly the set of all voters whose votes are $b \succ a$; similarly for "a"; "all" denotes the set of all voters. For example, if two voters vote for $a \succ b$ and three voters vote for $b \succ a$, then the level-3 pivotal set consists of exactly the two voters whose votes are $a \succ b$.*

# of $a \succ b$	Pivotal sets				
	1	2	3	4	...
0	$\emptyset$	$\emptyset$	all	$\emptyset$	...
1	$\emptyset$	b	all	b	...
2	b	all	a	all	...
3	a	all	b	all	...
4	$\emptyset$	a	all	a	...
5	$\emptyset$	$\emptyset$	all	$\emptyset$	...

Table 1: The pivotal sets under majority.

Proposition 1 *For any $k \in \mathbb{N}$, D_r^k is strategic-pivot-proof.*

Proof: For any $j \notin \bigcup_{i=1}^k \text{PS}_r^i(P)$ and any vote V_j', we prove that for any $i \leq k$, $j \notin \text{PS}_r^i(P_{-j}, V_j')$. For the sake of contradiction, let $i \leq k$ and V_j' be such that $j \in \text{PS}_r^i(P_{-j}, V_j')$. By the definition of PS_r^i, there exists a vote V_j^* such that $\text{PS}_r^{i-1}(P_{-j}, V_j') \neq \text{PS}_r^{i-1}(P_{-j}, V_j^*)$. Therefore, either $\text{PS}_r^{i-1}(P_{-j}, V_j') \neq \text{PS}_r^{i-1}(P)$ or $\text{PS}_r^{i-1}(P_{-j}, V_j^*) \neq \text{PS}_r^{i-1}(P)$. In both cases $j \in \text{PS}_r^i(P)$, which contradicts the assumption. □

3.2 Coalitional pivotal sets and Shapley-Shubik power index

When defining hierarchical pivotal sets, we are concerned with the voting power for a single voter to (indirectly) change the winner. It is natural to consider the voting power for a coalition of voters to change the winner by voting collaboratively. We first define the set of pivotal coalitions.

Given a profile P, a subset $S \subset N$ is a *pivotal coalition*, if there exists a profile P_S' for the voters in S such that $r(P) \neq r(P_{-S}, P_S')$. We define the indicator function v_r^P as follows. For any coalition $S \subseteq N$, if S is a pivotal coalition, then $v_r^P(S) = 1$; otherwise $v_r^P(S) = 0$. For any voting rule r and any profile P, let $\text{CPS}_r(P)$ denote the set of all pivotal coalitions, that is, $\text{CPS}_r(P) = \{S \subseteq N : v_r^P(S) = 1\}$. Obviously, if a set of voters S can change the winner, then any superset of S can also change the winner. Therefore, for any r and any profile P, $\text{CPS}_r(P)$ is *upward-closed*, that is, for any $S \in \text{CPS}_r(P)$ and any S' such that $S \subseteq S'$, we have $S' \in \text{CPS}_r(P)$.

Example 2 *There are three alternatives $\{a, b, c\}$. Let P=($a \succ b \succ c$, $a \succ c \succ b$, $c \succ a \succ b$). We have $CPS_{\text{Plu}}(P) = \{\{1\}, \{2\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$ and $CPS_{\text{Veto}}(P) = \{\{1\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.*

Now, a voter's voting power can be defined similarly to the Shapley-Shubik power index [13]. We now define a *power index* w_r that measures a voter's marginal contribution in making coalitions pivotal. Let $w_r : \text{F}_n(\mathcal{C}) \times N \to \mathbb{R}_{\geq 0}$ be a

mapping such that for any profile $P \in \mathrm{F}_n(\mathcal{C})$ and any $j \leq n$, we have:

$$w_r(P,j) = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(n-|S|-1)!}{n!}(v_r^P(S \cup \{j\}) - v_r^P(S))$$

Proposition 2 *For any rule r that does not always select the same alternative and any profile P, $\sum_{j=1}^{n} w_r(P,j) = 1$.*

To the best of our knowledge, this is the first time that Shapley-Shubik power index is considered in the context of preference aggregation by voting rules.

Based on the power index w_r, we define a weighted random dictatorship D_r^w as follows. If $\mathrm{CPS}_r(P) = \emptyset$ (or equivalently, r always selects the same alternative), then $D_r^w(P) = \emptyset$. Otherwise, for any profile P, $D_r^w(P)$ is the distribution over N that chooses j with probability $w_r(P,j)$.

Example 3 *Let P be the same profile as defined in Example 2. $D_{\mathrm{Plu}}^w(P)$ chooses 1 and 2 with the same probability $1/2$; $D_{\mathrm{Veto}}^w(P)$ chooses 1 with probability $2/3$, and chooses 2 and 3 with the same probability $1/6$.*

3.3 Relationships between the two pivotal sets

The next theorem states that the smallest k such that the level-k pivotal set is non-empty equals to the size of the smallest coalitional pivotal set for P.

Theorem 1 *For any voting rule r and any profile P,*

$$\min\{k : PS_r^k(P) \neq \emptyset\} = \min_{S \in CPS_r(P)}\{|S|\}$$

Proof: Let $k^* = \arg\min_k\{\mathrm{PS}_r^k(P)\}$ and $k' = \min_{S \in \mathrm{CPS}_r(P)}\{|S|\}$. We first prove that $k^* \leq k'$. Suppose for the sake of contradiction that $k^* > k'$. Without loss of generality, $S = \{1, \ldots, k'\}$, and let $P'_S = (V'_1, \ldots, V'_{k'})$ be the votes such that $r(P) \neq r(P_{-S}, P'_S)$. For any $k \leq k'$, let $P_k = (V'_1, \ldots, V'_k, V_{k+1}, \ldots, V_n)$, that is, P_k is obtained from P by replacing the first k votes by $V'_1, \ldots, V'_k$, respectively. Because $k' < k^*$, for any $k \leq k'$, $\mathrm{PS}_r^k(P) = \emptyset$. Therefore, for any $k \leq k' - 1$, changing the vote of voter 1 from V_1 to V'_1 does not change the level-k pivotal set. That is, for any $k \leq k' - 1$, $\mathrm{PS}_r^k(P_1) = \emptyset$. Similarly, it is easy to see that for any $i \leq k' - 1$, for any $k \leq k' - i$, $\mathrm{PS}_r^k(P_i) = \emptyset$. Specifically, $\mathrm{PS}_r^1(P_{k'-1}) = \emptyset$. It follows from $\mathrm{PS}_r^1(P) = \emptyset$ and for any $i \leq k' - 1$, $\mathrm{PS}_r^1(P_i) = \emptyset$, that $r(P) = r(P_1) = r(P_2) = \ldots = r(P_{k'})$. This contradicts the assumption that $r(P) \neq r(P_{k'})$. Consequently, $k^* \leq k'$.

Next, we prove that $k' \leq k^*$. It suffices to prove that for any $k \leq k' - 1$, $\mathrm{PS}_r^k(P) = \emptyset$. We have following stronger claim, whose proof is omitted due to the space constraint.

Claim 1 *For any $2 \leq q \leq k'$, any P' that differs from P on no more than $k' - q$ votes, and any $k \leq q - 1$, $PS_r^k(P') = \emptyset$.*

Let $q = k'$ in Claim 1, we have that $\mathrm{PS}_r^{k'-1}(P) = \emptyset$, which means that $k^* \geq k'$. Therefore, $k^* = k'$. □

4 Hierarchical pivotal sets for anonymous voting rules

In the remainder of the paper, we focus on hierarchical pivotal sets. It is easy to see that if a voter is not pivotal in *any* profile, then for any k and any profile P, she is not in the level-k pivotal set. Such a voter is said to be *redundant*.

Definition 3 *Given a voting rule r, a voter j is* redundant*, if for any profile P and any vote V'_j, $r(P) = r(P_{-j}, V'_j)$.*

If a voter is redundant, then effectively her vote can be completely ignored. Therefore, for any profile, none of the voters in the union of its hierarchical pivotal sets (as $k \to \infty$) is redundant. That is, the union of the hierarchical pivotal sets for any profile is a *sound* characterization of the non-redundant voters. We ask the following two natural questions. The first question asks whether or not the union of the hierarchical pivotal sets for a given profile P is a *complete* characterization of the non-redundant voters.

Question 1 *Given a voting rule r, is it true that for any non-redundant voter j and any profile P, there exists $k \in \mathbb{N}$ such that j is in the level-k pivotal set for P?*

The second question concerns the asymptotic property of level-k pivotal sets when k goes to infinity. Given a profile P, we are asked whether the level-k pivotal sets for P will converge (to the empty set), when k goes to infinity.

Question 2 *Given a voting rule r, does there exist $K \in \mathbb{N}$ such that for any $k \geq K$, the level k-pivotal set is $\emptyset$?*

In this section, we give an affirmative answer to Question 1 for any voting rule that satisfies anonymity and unanimity, and a negative answer to Question 2 for the majority rule. We first prove a lemma, which states that for any anonymous voting rule r, if a voter j is in the level-k pivotal set for a profile P, then other voters who cast the same vote as j's vote are also in the level-k pivotal set for P. This lemma will be frequently used in this paper. Due to the space constraint, some proofs are omitted.

Lemma 1 *For any anonymous voting rule r, any profile P, any $k \in \mathbb{N}$, and any pair of voters i, j with $V_i = V_j$, $i \in PS_r^k(P)$ if and only if $j \in PS_r^k(P)$.*

Lemma 1 states that for any anonymous voting rule r and any profile P, a voter's membership in the level-k pivotal set can be characterized by her vote. Therefore, for any anonymous voting rule r and any profile, the level-k pivotal set can be represented by the set of all votes that are cast by some level-k pivotal voters. We will use this observation later in the paper, especially in Section 6. The next theorem gives an affirmative answer to Question 1 for any voting rule that satisfies anonymity and unanimity.

Theorem 2 *Let r be a voting rule that satisfies anonymity and unanimity. For any n-profile P and any voter j, there exists $k \leq \min_{S \in CPS_r(P)}\{|S|\} + 1 \leq n + 1$ such that $j \in PS_r^k(P)$.*

Proof: Let $K = \min_{S \in \mathrm{CPS}_r(P)}\{|S|\}$. For the sake of contradiction, without loss of generality for any $k \leq K + 1$, $1 \notin \mathrm{PS}_r^k(P)$. By Theorem 1, there exists $k^* \leq K$ such that $\mathrm{PS}_r^{k^*}(P) \neq \emptyset$. Let $j^* \in \mathrm{PS}_r^{k^*}(P)$ and W be the vote of voter j^*. Let $P' = (P_{-1}, W)$, that is, P' is the profile obtained from P by letting voter 1 vote for W. Because $1 \notin \mathrm{PS}_r^{k^*}(P)$ and $1 \notin \mathrm{PS}_r^{k^*+1}(P)$, we have that $1 \notin \mathrm{PS}_r^{k^*}(P')$. It follows from Lemma 1 that for any voter j whose vote is W in P', $j \notin \mathrm{PS}_r^{k^*}(P')$. Specifically, $j^* \notin \mathrm{PS}_r^{k^*}(P')$, which means that $\mathrm{PS}_r^{k^*}(P') \neq \mathrm{PS}_r^{k^*}(P)$. Therefore, $1 \in \mathrm{PS}_r^{k^*+1}(P)$. This contracts the assumption that $1 \notin \mathrm{PS}_r^{k^*+1}(P)$. □

Theorem 2 is quite positive. It implies that if we remove all redundant voters, D_r^k can be used to approximate the fully random dictatorship, which is strategy-proof. It is a very interesting topic to study how good this approximation is, which we left as an open problem.

For Question 2, suppose the level-k pivotal set converges as k goes to infinity, we first prove that it must converge to $\emptyset$.

Proposition 3 *For any anonymous voting rule r, if there exists k such that for any n-profile P, $PS_r^k(P) = PS_r^{k+1}(P)$, then for any n-profile P, $PS_r^k(P) = \emptyset$.*

However, Proposition 3 does not guarantee the existence of k such that $\mathrm{PS}_r^k(P) = \mathrm{PS}_r^{k+1}(P)$. In fact, the next proposition shows that such a k might not exist for the majority rule, which satisfies anonymity and unanimity. Therefore, the answer to Question 2 is negative.

Proposition 4 *Let there be two alternatives $\{a, b\}$, 5 voters, and we use the majority rule. There does not exist $k \in \mathbb{N}$ such that for any profile P, the level-k pivotal set for P is $\emptyset$.*

Proof: From Table 1 in Example 1, it is easy to see that for any profile, its level-2 and level-4 pivotal sets are identical and are different from level-3 pivotal sets. Therefore, for any profile, none of the level-k pivotal sets converges as k goes to infinity. □

5 Hierarchical pivotal sets for non-anonymous voting rules

In this section, we focus on non-anonymous voting rules. Surprisingly, for some voting rules that do not satisfy anonymity, the answer to Question 1 is negative.

Proposition 5 *Let $m = 4$ and $n = 3$. There exists a non-anonymous voting rule r that satisfies the following conditions.*

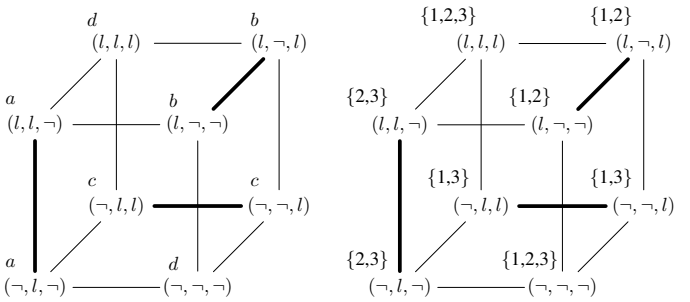
- *No voter is redundant.*
- *For any $k \in \mathbb{N}$ and any profile P such that $|P| = 3$, the level-k pivotal set for P is non-empty.*
- *For any voter j, there exists a profile P such that $|P| = 3$ and for any $k \in \mathbb{N}$, j is not in the level-k pivotal set for P.*

Proof: Let the four alternatives be $\{a, b, c, d\}$. Let $l = [a \succ b \succ c \succ d]$. We define a voting rule r as follows. $r(l, l, \neg) = r(\neg, l, \neg) = a$, $r(l, \neg, l) = r(l, \neg, \neg) = b$, $r(\neg, l, l) = r(\neg, \neg, l) = c$, $r(l, l, l) = r(\neg, \neg, \neg) = d$.

Here "$\neg$" means any linear order that is different from l. For example, $r(\neg, l, l) = c$ means that for any 3-profile where voter 1's voter is not l, and the votes of voter 2 voter and 3 are both l, the winner is c. The voting rule is illustrated in Figure 1(a), where each vertex represents a set of 3-profiles and the alternative associated with it is the winner for these profiles. An edge between two vertices A and B in the graph means that for any profile P in A, there exists a profile P' in B such that P' can be obtained from P by changing exactly one vote. An edge is bold if the winners for its two endpoints are the same. We have the following claim (whose proof is omitted due to the space constraint.)

Claim 2 *For any $k \in \mathbb{N}$ and any profile P, $PS_r^k(P) = PS_r^{k+1}(P)$, and is illustrated in Figure 1 (b).*

It follows from Claim 2 that r satisfies all the properties in the description of the proposition. □



(a) The voting rule r. (b) The level-k pivotal set for any k.

Figure 1: The voting rule r and the hierarchical pivotal sets.

6 Computing hierarchical pivotal sets

In this section, we investigate the computational complexity of computing level-k pivotal sets. We first relate the problem of computing level-1 pivotal sets to the *unweighted coalitional manipulation (UCM)* problems with a single manipulator. An instance of UCM is a tuple (r, P^{NM}, c, M), where r is a voting rule, P^{NM} is the non-manipulators' profile, c is the manipulators' preferable alternative, and M is the set of manipulators. We are asked whether there exists a profile P^M for the manipulators such that $r(P^{NM} \cup P^M) = c$. Let UCM_1 denote the UCM problems with a single manipulator, that is, $|M| = 1$.

Proposition 6 *For any voting rule r, if UCM_1 is in P, then computing $PS_r^1(P)$ is also in P.*

Following the results of computing UCM_1 for common voting rules [3; 2; 4; 7; 16; 15], we immediately obtain the following corollary.

Corollary 1 *For any $r \in \{$Copeland, Veto, Plurality with runoff, Cup, Maximin, Bucklin, Borda$\}$ and any profile P, there exists a polynomial-time algorithm that computes $PS_r^1(P)$.*[3]

For STV and ranked pairs, UCM_1 is NP-complete [2; 15]. The next two theorems show that computing the level-1 pivotal sets for them are NP-complete. It is not clear whether there exists a general reduction that works for any voting rule.

Theorem 3 *It is NP-complete to compute $PS_r^1(P)$ for r=STV.*

Proof: It is easy to check that computing $\mathrm{PS}_r^1(P)$ for STV is in NP. We prove the NP-hardness by a reduction from a special kind of UCM_1 problems for STV, where c is ranked in the top position in at least one vote in P^{NM}. This problem has been shown to be NP-complete [2]. For any UCM_1 instance $(\mathrm{STV}, P^{NM}, c, \{n\})$ where c is ranked in the top position in at least one vote in P^{NM} ($|P^{NM}| = n - 1$), we construct the following instance of computing the level-1 pivotal set. Let $\mathcal{C}$ denote the set of alternatives in the UCM_1 instance.

Alternatives: $\mathcal{C} \cup \{d\}$, where d is an auxiliary alternative.

Profile: Let P denote a profile of $2n - 1$ votes as follows. The first $n-1$ votes are obtained from P^{NM} by putting d right below c. The next n votes ranks d in the first position (other alternatives are ranked arbitrarily). We are asked whether $n \in \mathrm{PS}_{\mathrm{STV}}^1(P)$.

[3]The definition of these voting rules can be found in e.g. [15].

It is easy to check that $\text{STV}(P) = d$. Suppose the UCM_1 instance has a solution, denoted by V. Then, let V' denote the linear order over $\mathcal{C} \cup \{d\}$ obtained from V by ranking d in the bottom position. Let P' denote the profile where voter n changes her vote to V'. We note that d is ranked in the top position for $n-1$ time in P'. Therefore, d is never eliminated in the first $|\mathcal{C}| - 1$ rounds. Moreover, for any $j \leq |\mathcal{C}| - 1$, the alternative that is eliminated in the jth round for P' is exactly the same as the alternative that is eliminated in the jth round for P. In the lasts round, c is ranked in the top position for n time, which means that $\text{STV}(P') = c \neq d$. Hence, $n \in \text{PS}^1_{\text{STV}}(P)$.

On the other hand, if $n \in \text{PS}^1_{\text{STV}}(P)$, then there exist a vote V' such that by changing her vote to V', voter n can change the winner under STV. Let $P' = (P_{-n}, V')$. Again, because d is ranked in the top position for at least $n-1$ time in P', it will only be eliminated in the last round. We recall that c is ranked in the first position in at least one vote in P^{NM}, and d is ranked right below c in the corresponding vote in P'. Therefore, d beats all alternatives in $\mathcal{C} \setminus \{c\}$ in their pairwise elections, which means that in the last round the only remaining alternatives must be c and d. Let V be a linear order obtained from V' by removing d. It follows that V is a solution to the UCM_1 instance.

Therefore, computing the level-1 pivotal set for STV is NP-complete. □

Theorem 4 (proof omitted due to the space constraint) *It is NP-complete to compute $PS^1_r(P)$ for r=RP (ranked pairs).*

For any anonymous voting rule, when m is bounded above by a constant, we can find a dynamic-programming algorithm that computes the level-k pivotal set. The algorithm is based on the following two key observations. First, when the number of alternatives is bounded above by a constant, the number of essentially different profiles is polynomial. Second, by Lemma 1, a level-k pivotal set can be represented succinctly by a set of votes (instead of voters). The details of the algorithm is omitted due to the space constraint.

7 Future research

There are many interesting directions for future research. For example, in this paper we have three open problems. How can we compare the strategic-pivot-proofness and strategy-proofness? How good/bad it is to use D^k_r to approximate the fully random dictatorship? What is the computational complexity of computing level-k ($k \geq 2$) pivotal sets for common voting rules? Moreover, we believe that defining and computing voting power in the traditional voting setting (in contrast to the weighted voting games) is an important topic. It would be worthwhile studying applications of the two types of voting powers proposed in this paper (especially the Shapley-Shubik power index), for example, in defining other (weighted) random dictatorships or in the coalition formation of the manipulators. Besides these topics, we can definitely examine other ways of defining voting power, for example by using the Banzhaf power index [1].

Acknowledgements

Lirong Xia acknowledges a James B. Duke Fellowship and Vincent Conitzer's NSF CAREER 0953756 and IIS-0812113, and an Alfred P. Sloan fellowship for support. We thank all anonymous IJCAI-11 and WSCAI reviewers for their helpful suggestions and comments.

References

[1] John F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965.

[2] John Bartholdi, III and James Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[3] John Bartholdi, III, Craig Tovey, and Michael Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[4] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3): 1–33, 2007.

[5] Anthony Downs. *An Economic Theory of Democracy*. New York: Harper & Row, 1957.

[6] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53:74–82, 2010.

[7] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Copeland voting: ties matter. In *Proc. of AAMAS*, pages 983–990, 2008.

[8] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[9] Timothy J. Feddersen. Rational choice theory and the paradox of not voting. *JEP*, 18(1):99–112, 2004.

[10] Benny Geys. 'Rational' theories of voter turnout: A review. *Political Studies Review*, 4(1):16–35, 2006.

[11] Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45:665–681, 1977.

[12] Ariel D. Procaccia and Jeffrey S. Rosenschein. Average-case tractability of manipulation in voting via the fraction of manipulators. In *Proc. AAMAS*, pages 718–720, 2007.

[13] Lloyd Shapley and Martin Shubik. A method for evaluating the distribution of power in a committee system. *The American Political Science Review*, 48(3):787–792, 1954.

[14] Lirong Xia and Vincent Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proc. EC*, pages 109–118, 2008.

[15] Lirong Xia, Michael Zuckerman, Ariel D. Procaccia, Vincent Conitzer, and Jeffrey Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proc. IJCAI*, pages 348–353, 2009.

[16] Michael Zuckerman, Ariel D. Procaccia, and Jeffrey S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412, 2009.

Weights in stable marriage problems increase manipulation opportunities

Maria Silvia Pini[1], **Francesca Rossi**[1], **Kristen Brent Venable**[1], **Toby Walsh**[2]
[1] : Department of Pure and Applied Mathematics, University of Padova, Italy.
Email: {mpini,frossi,kvenable}@math.unipd.it.
[2] : NICTA and UNSW, Sydney, Australia. Email: tw@cse.unsw.edu.au

Abstract

The stable marriage problem is a well-known problem of matching men to women so that no man and woman, who are not married to each other, both prefer each other. Such a problem has a wide variety of practical applications, ranging from matching resident doctors to hospitals, to matching students to schools or more generally to any two-sided market. In the classical stable marriage problem, both men and women express a strict preference order over the members of the other sex, in a qualitative way. Here we consider stable marriage problems with weighted preferences: each man (resp., woman) provides a score for each woman (resp., man). In this context, we consider the manipulability properties of the procedures that return stable marriages. While we know that all procedures are manipulable by modifying the preference lists or by truncating them, here we consider if manipulation can occur also by just modifying the weights while preserving the ordering and avoiding truncation. It turns out that, by adding weights, we indeed increase the possibility of manipulating and this cannot be avoided by any reasonable restriction on the weights.

1 Introduction

The stable marriage problem (SM) [3] is a well-known problem of matching the elements of two sets. It is called the *stable marriage* problem since the standard formulation is in terms of men and women, and the matching is interpreted in terms of a set of marriages. Given n men and n women, where each person expresses a strict ordering over the members of the opposite sex, the problem is to match the men to the women so that there are no two people of opposite sex who would both rather be matched with each other than with their current partners. If there are no such people, all the marriages are said to be *stable*. In [1] Gale and Shapley proved that it is always possible to find a matching that makes all marriages stable, and provided a polynomial time algorithm which can be used to find one of two extreme stable marriages, the so-called *male-optimal* or *female-optimal* solutions. The Gale-Shapley algorithm has been used in many real-life scenarios [12], such as in matching hospitals to resident doctors [6], medical students to hospitals, sailors to ships [8], primary school students to secondary schools [13], as well as in market trading.

In the classical stable marriage problem, both men and women express a strict preference order over the members of the other sex in a qualitative way. Here we consider stable marriage problems with weighted preferences (SMWs). In such problems, each man (resp., woman) provides a score for each woman (resp., man). Stable marriage problems with weighted preferences are more general than classical stable marriage problems. Moreover, they are useful in some real-life situations where it is more natural to express scores, that can model notions such as profit or cost, rather than a qualitative preference ordering.

In [10] we have defined new notions of stability for SMWs which depend on the scores given by the agents. In this paper, we study if the stable marriage procedures which return one of these new stable marriages are manipulable. In [11] Roth has shown that, when there are at least three men and three women, every stable marriage procedure is manipulable, i.e., there is a profile in which an agent can mis-report his preferences and obtain a stable marriage which is better than or equal to the one obtained by telling the truth. In this setting, mis-reporting preferences means changing the preference ordering [11] or truncating the preference list [2].

In this paper, we consider a possible additional way of misreporting one's own preferences, which is by just modifying the weights, in a way such that the orderings are preserved and the lists remain complete. We show that it is actually possible to manipulate by just doing this. Thus adding weights makes stable marriage procedures less resistant to manipulation. Moreover, we show that there are no reasonable restrictions on the weights that can prevent such manipulation.

2 Stable marriage problems with weighted preferences

A *stable marriage problem* (SM) [3] of size n is the problem of finding a stable matching between n men and n women. The men and women each have a preference ordering over the members of the other sex. A matching is a one-to-one correspondence between men and women. Given a matching M, a man m, and a woman w, the pair (m, w) is a *blocking*

pair for M if m prefers w to his partner in M and w prefers m to her partner in M. A matching is said to be *stable* if it does not contain blocking pairs. The sequence of preference orderings of all the men and women is called a *profile*. In the case of the classical stable marriage problem (SM), a profile is a sequence of strict total orders. Given a SM P, there may be many stable matchings for P, and always at least one. The *Gale-Shapley* (GS) *algorithm* [1] is a well-known algorithm that finds a stable matching in polynomial time. Given any procedure f to find a stable matching for an SM problem P, we will denote by $f(P)$ the matching returned by f.

Example 1 *Assume $n = 3$ and let $\{w_1, w_2, w_3\}$ and $\{m_1, m_2, m_3\}$ be respectively the set of women and men. The following sequence of strict total orders defines a profile: $\{m_1 : w_1 > w_2 > w_3$ (i.e., man m_1 prefers woman w_1 to w_2 to w_3); $m_2 : w_2 > w_1 > w_3$; $m_3 : w_3 > w_2 > w_1; w_1 : m_1 > m_2 > m_3$; $w_2 : m_3 > m_1 > m_2$; $w_3 : m_2 > m_1 > m_3$. This profile has two stable matchings: the male-optimal solution which is $\{(m_1, w_1), (m_2, w_2), (m_3, w_3)\}$ and the female-optimal which is $\{(m_1, w_1), (m_2, w_3), (m_3, w_2)\}$.* □

In SMs, each preference ordering is a strict total order over the members of the other sex. More general notions of SMs allow preference orderings to have ties [9; 5; 4]. We will denote with SMT a *stable marriage problem with ties*. A matching M for a SMT is said to be *weakly-stable* if it does not contain blocking pairs. Given a man m and a woman w, the pair (m, w) is a blocking pair for M if m and w are not married to each other in M and each one *strictly prefers* the other to his/her current partner.

A *stable marriage problem with weighted preferences* (SMW) [7] is like a classical SM except that every man/woman gives also a numerical preference value for every member of the other sex, that represents how much he/she prefers such a person. Such preference values are natural numbers and higher preference values denote a more preferred item. The *preference value* for man m (resp., woman w) of woman w (resp., man m) will be denoted by $p(m, w)$ (resp., $p(w, m)$).

Example 2 *Let $\{w_1, w_2\}$ and $\{m_1, m_2\}$ be respectively the set of women and men. An instance of an SMW is the following: $\{m_1 : w_1^{[9]} > w_2^{[1]}$ (i.e., man m_1 prefers woman w_1 to woman w_2, and he prefers w_1 with weight 9 and w_2 with weight 1), $m_2 : w_1^{[3]} > w_2^{[2]}$, $w_1 : m_2^{[2]} > m_1^{[1]}$, $w_2 : m_1^{[3]} > m_2^{[1]}\}$.* □

In [10] we defined two notions of stability for SMWs based on weights. The first one is a simple generalization of the classical notion of stability: a blocking pair is a man and a woman that each prefer to be married to each other more than α with respect to being married to their current partner.

Definition 1 (α-stability) *Let us consider a natural number α with $\alpha \geq 1$. Given a matching M, a man m, and a woman w, the pair (m, w) is an α-blocking pair for M if m prefers w to his partner in M, say w', by at least α (i.e., $p(m, w) - p(m, w') \geq \alpha$), and w prefers m to her partner in M, say m', by at least α (i.e., $p(w, m) - p(w, m') \geq \alpha$). A matching is α-stable if it does not contain α-blocking pairs.*

In Example 2, if $\alpha = 1$, the only α-stable matching is $\{(m_1, w_2), (m_2, w_1)\}$. If instead $\alpha \geq 2$, then all matchings are α-stable.

To find an α-stable matching, it is useful to relate the α-stable matchings of an SMW to the stable matchings of a suitable classical stable marriage problem, so we can use classical stable marriage procedures. Given an SMW P, let us denote with $c(P)$ the classical SM problem obtained from P by considering only the preference orderings induced by the weights of P. If α is equal to 1, then the α-stable matchings of P coincide with the stable matchings of $c(P)$. In general, α-stability gives us more matchings that are stable, since we have a stronger notion of blocking pair. If we denote with $\alpha(P)$ the SMT obtained from an SMW P by setting as indifferent every pair of people whose weight differ for less than α, the α-stable matchings of P coincide with the weakly stable matchings of $\alpha(P)$.

The second notion of stability based on the weights, defined in [10], considers the happiness of a whole pair (a man and a woman) rather than that of each single person in the pair. Thus this notion depends on what we call the strength of a pair, rather than the preferences of each of two members of the pair.

Definition 2 (link-additive stability) *Given a man m and a woman w, the* link-additive strength *of the pair (m, w), denoted by $la(m, w)$, is the value obtained by summing the weight that m gives to w and the weight that w gives to m, i.e., $la(m, w) = p(m, w) + p(w, m)$. Given a matching M, the* link-additive value *of M, denoted by $la(M)$, is the sum of the links of all its pairs, i.e., $\sum_{\{(m,w)\in M\}} la(m, w)$. Given a matching M, a man m, and a woman w, the pair (m, w) is a* link-additive blocking pair *for M if $la(m, w) > la(m', w)$ and $la(m, w) > la(m, w')$, where m' is the partner of w in M and w' is the partner of m in M. A matching is* link-additive stable *if it does not contain any link-additive blocking pair.*

If we consider again Example 2, the pair (m_1, w_1) has link-additive strength equal to 10 (that is, 9+1), while pair (m_2, w_2) has strength 3 (that is, 2+1). The matching $\{(m_1, w_1), (m_2, w_2)\}$ has link-additive value 13 and it is link-additive stable. The other matching is not link-additive stable, since (m_1, w_1) is a link-additive blocking pair.

The reason why we used the terminology *link-additive* is that we compute the strength of a pair, as well as the value of a matching, by using the sum. However, we could use other operators, such as the maximum or the product. If we use the maximum, we will use *link-max* instead of *link-additive*.

Again, we can relate the link-additive (resp., link-max) stable matchings of an SMW to the stable matchings of a suitable classical SM problem. Given an SMW P, let us denote with $Linka(P)$ (resp., $Linkm(P)$) the stable marriage problem with ties obtained from P by taking the preference orderings induced by the link-additive (link-max) strengths of the pairs. Then, a matching is link-additive (resp., link-max) stable iff it is a weakly stable matching of $Linka(P)$. An optimal link-additive (resp. link-max) stable matching is one with maximal link-additive (resp., link-max) value.

3 W-manipulation

We know that, with at least three men and three women, every stable marriage procedure is manipulable [11], i.e., there is a profile where an agent, mis-reporting his preferences, obtains a stable matching which is better than the one obtained by telling the truth. In stable marriage problems, agents can try to manipulate in two ways: by changing the preference ordering [11], or by truncating the preference list [2].

In SMW problems, there is another way of lying: changing the weights. We show this gives the agents an additional power to manipulate even if the manipulator just changes the weights, while preserving the preference ordering and does not truncate the preference list.

A stable marriage procedure f is *w-manipulable* (resp., *strictly w-manipulable*) if there is a pair of profiles p and p' that contain the same preference orderings but differ in the weights of an agent, say w, such that $f(p')$ is better than or equal to (resp., better than) $f(p)$ for w.

4 W-manipulation for α-stability

We first assume that the agents know the value of α.

Theorem 1 *Let α be any natural number > 1. Every procedure which returns an α-stable matching is w-manipulable, and there is at least one procedure which is strictly w-manipulable.*

Proof: Let $\{w_1, w_2\}$ and $\{m_1, m_2\}$ be, respectively, the set of women and men. Consider the following instance of an SMW, say P, $\{m_1 : w_1^{[x+\alpha]} > w_2^{[x]}, m_2 : w_1^{[x+\alpha]} > w_2^{[x]}, w_1 : m_1^{[x+\alpha]} > m_2^{[x+1]}, w_2 : m_1^{[x+\alpha]} > m_2^{[x]}\}$, where x is any value greater than 0. P has two α-stable matchings: $M_1 = \{(m_1, w_1), (m_2, w_2)\}$ and $M_2 = \{(m_1, w_2), (m_2, w_1)\}$. Assume that w_1 mis-reports her preferences as follows: $w_1 : m_1^{[x+\alpha]} > m_2^{[x]}$, i.e., assume that she changes the weight given to m_2 from $x+1$ to x. Let us denote with P' the resulting problem. P' has a unique α-stable matching, that is M_1, which is the best α-stable matching for w_1 in P. Therefore, it is possible for w_1 to change her weights to get a better or equal result than the one obtained by telling the truth. Also, since P' has a unique α-stable matching, every procedure which returns an α-stable matching returns such a matching. Thus, every procedure is w-manipulable. Moreover, if we take the procedure which returns M_2 in the first profile, this example shows that this procedure is strictly w-manipulable. □

Thus, when using weights, agents can manipulate by just modifying the weights, if they know which α will be used.

Let us now see whether there is any syntactical restriction over the profiles that can prevent this additional form of manipulation. First, we may notice that this manipulation is only related to the fact that some distances between adjacent weights are made larger or smaller. This, depending on the chosen α, may imply that some elements are considered in a tie or ordered in $\alpha(P)$. Thus, a manipulator may introduce a tie that was not in its real preference ordering, or may eliminate a tie from this ordering. Based on this consideration, we can consider restricting our attention to profiles where ties are not allowed. But this would simply mean eliminating the weights, since in this case the α-stable matchings would coincide with the stable matchings of the SM obtained by just forgetting the weights. We can thus consider what happens if we allow at most one tie (that is, a difference less than α) in each preference ordering. Even this strong restriction does not avoid w-manipulation, since the example in the proof of Theorem 1 respects this restriction. A weaker restriction would be to allow at most one tie in the whole profile, but this would mean requiring coordination between the agents or knowing who is the manipulator. Also, again the same example obeys this restriction. Summarizing, if agents know the value of α, there is no way to prevent w-manipulation!

Some hope remains for when α is not known by the agents. Assume that this is the case, but agents know that α is bounded by a certain value, say α_{max}. Unfortunately, again the example in the proof of Theorem 1 (where we replace every α with α_{max}) holds. Thus every procedure is still w-manipulable, and some are also strictly w-manipulable. Also, restricting to at most one tie per agent will not avoid w-manipulation, since again the same example holds.

The most promising case is when agents have no information about α. In this case, we need to define what it means for a procedure to be manipulable: a procedure which returns an α-stable matching is *α-w-manipulable* if it is w-manipulable for all α and it is strictly w-manipulable for at least one α.

Theorem 2 *There is a procedure which returns an α-stable matching which is α-w-manipulable.*

Proof: Let $\{w_1, w_2\}$ and $\{m_1, m_2\}$ be, respectively, the set of women and men. Consider the following instance of an SMW, P, $\{m_1 : w_1^{[3]} > w_2^{[2]}, m_2 : w_2^{[3]} > w_1^{[2]}, w_1 : m_2^{[3]} > m_1^{[2]}, w_2 : m_1^{[3]} > m_2^{[2]}\}$. For every α, P has two α-stable matchings: $M_1 = \{(m_1, w_1), (m_2, w_2)\}$ and $M_2 = \{(m_1, w_2), (m_2, w_1)\}$. When $\alpha = 1$, M_2 is strictly better than M_1 for w_1 in P, while when $\alpha > 1$, M_2 is equally preferred to M_1 for w_1 in P.

Assume that w_1 mis-reports her preferences as follows: $w_1 : m_2^{[3]} > m_1^{[1]}$. Let us denote with P' the problem obtained from P by using this mis-reported preference for w_1. When $\alpha \in \{1, 2\}$, M_2 is strictly better than M_1 for w_1 in P', while when $\alpha > 2$, M_2 is equally preferred to M_1 for w_1 in P'.

Let us consider a procedure, that we call mGS, which works as the Gale-Shapley algorithm over all the profiles except on P and P', where it works as follows: if a matching is strictly better than another matching in terms of α for w_1, then it returns the best one, while if a matching is equally preferred to another matching in terms of α for w_1, then it returns the worst one for w_1 w.r.t. the strict preference ordering induced by the weights. Therefore, when $\alpha = 1$, mGS returns M_2 in both P and P', when $\alpha = 2$ mGS returns M_1 in P and M_2 in P', while when $\alpha > 2$ mGS returns M_1 in both P and P'. Therefore, if w_1 lies, for every α, he obtains a partner that is better than or equal to the one obtained by telling the truth, and there is a value α (i.e., α=2) where he obtains a partner that is better than the one obtained by telling the truth. Therefore, the mGS procedure is α-w-manipulable. □

As in the case when α is known, we may consider restricting to profiles with at most one tie per agent. However, the example in the above proof satisfies this restriction, so it shows that α-w-manipulability is possible also with such a severe restriction.

Summarizing, in the context of α-stability, no matter whether we have information about α or not, it is possible to have w-manipulability, even if we severely restrict the profiles.

5 W-manipulation for link-additive stability

We next show that every procedure for link-additive stability is strictly w-manipulable.

Theorem 3 *Every procedure that returns a link-additive stable matching is strictly w-manipulable.*

Proof: Let $\{w_1, w_2\}$ and $\{m_1, m_2\}$ be, respectively, the set of women and men. Consider the following instance of an SMW, say P: $\{m_1 : w_2^{[6]} > w_1^{[4]}, m_2 : w_2^{[5]} > w_1^{[4]}, w_1 : m_1^{[4]} > m_2^{[3]}, w_2 : m_1^{[3]} > m_2^{[2]}\}$. P has a unique link-additive stable matching, which is $M_1 = \{(m_1, w_2), (m_2, w_1)\}$. Assume that w_1 mis-reports her preferences as follows: $w_1 : m_1^{[5000]} > m_2^{[2]}$. Then, in the new problem, that we call P', there is only one stable matching, which is $M_2 = \{(m_1, w_1), (m_2, w_2)\}$, and M_2 is better than M_1 for w_1 in P. Since there is only one stable matching in both P and P', every procedure which returns a link-additive stable matching will return M_2 in P and M_1 in P', and thus it is strictly w-manipulable. □

The example in the proof of the above theorem shows a very intuitive and dangerous manipulation scheme: the manipulator sets a very high weight (higher than twice the highest of the other weights in the profile) for its top choice. In this way, it will surely be matched to its top choice, no matter the procedure used or the preferences of the other agents over the alternatives that are not their top choices.

This form of manipulation can be avoided by forcing the same weight for all top choices of all agents. This restriction however does not prevent all forms of w-manipulation.

Theorem 4 *If we restrict to profiles with the same weight for all top choices, every procedure that returns a link-additive stable matching is w-manipulable, and there is at least one procedure which is strictly w-manipulable.*

Proof: Let $\{w_1, w_2, w_3\}$ and $\{m_1, m_2, m_3\}$ be, respectively, the set of women and men. Consider the following instance of an SMW, P, $\{m_1 : w_3^{[7]} > w_2^{[6]} > w_1^{[5]}, m_2 : w_3^{[7]} > w_2^{[6]} > w_1^{[5]}, m_3 : w_3^{[7]} > w_2^{[6]} > w_1^{[5]}, w_1 : m_3^{[7]} > m_1^{[5]} > m_2^{[4]}, w_2 : m_3^{[7]} > m_1^{[5]} > m_2^{[4]}, w_3 : m_3^{[7]} > m_1^{[6]} > m_2^{[5]}\}$. P has an unique link-additive stable matching, which is $M_1 = \{(m_1, w_2), (m_2, w_1), (m_3, w_3)\}$. Assume that w_1 mis-reports her preferences as follows: $w_1 : m_3^{[7]} > m_1^{[6]} > m_2^{[4]}$. Then, in the new problem, that we call P', there are only two link-additive stable matchings, i.e., M_1 and $M_2 = \{(m_1, w_1), (m_2, w_2), (m_3, w_3)\}$, where M_2 is better than M_1 for w_1. Thus every procedure is w-manipulable. If we consider the procedure that returns matching M_2 in P', this pair of profiles shows that this procedure is strictly w-manipulable. □

Notice that, if we consider profiles where all top choices have the same weight and all differences (of weights of adjacent items in the preference lists) are exactly 1, then weights are fixed and are thus irrelevant. Also, obviously w-manipulation cannot occur, since agents cannot modify the weights. We may wonder whether, by restricting to profiles which are close to this extreme case, we may avoid w-manipulation. Unfortunately, this is not so. In fact, we can consider just profiles with the same weight for all top choices and where at most one difference is 2, while all the others are 1, for every agent. This holds for the example in the proof of Theorem 4. This shows that even this strong restriction is not enough to avoid w-manipulation.

If we restrict our attention to procedures that return optimal link-additive or link-max stable matchings, we can still prove that all such procedures are strictly w-manipulable, and they are w-manipulable when all top choices have the same weight. In fact, the same examples in the proofs of Theorem 3 and 4 still hold.

6 Conclusions and future work

We have investigated the manipulation properties of stable marriage problems with weighted preferences, and considered two different notions of stability. We have shown that, in both cases, adding weights to classical stable marriage problems increases the possibility of manipulating the resulting matching, since agents can manipulate even by just modifying the weights, without changing or truncating the preference lists. We have also shown that reasonable restrictions over the weights do not avoid such additional forms of manipulation. However, in the case of link-additive stability, forcing all top choices to have the same weight for all agents prevents an extreme form of w-manipulation, which would allow the manipulator to dictate its own partner in every link-additive stable matching.

We plan to investigate the computational complexity of w-manipulation. We also plan to use scoring-based voting rules to choose among the stable matchings, and to adapt existing results about manipulation complexity for such voting rules to weighted stable marriage problems.

References

[1] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *Amer. Math. Monthly*, 69:9–14, 1962.

[2] D. Gale and M. Sotomayor. Semiring-based constraint solving and optimization. *American Mathematical Monthly*, 92:261–268, 1985.

[3] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Boston, Mass., 1989.

[4] M. Halldorsson, R. W. Irving, K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, and S. Scott. Approximability results for stable marriage problems with ties. *Theor. Comput. Sci*, 306(1-3):431–447, 2003.

[5] R. W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48:261–272, 1994.

[6] R. W. Irving. Matching medical students to pairs of hospitals: a new variation on an old theme. In *Proc. ESA'98*, volume 1461 of *LNCS*, pages 381–392. Springer-Verlag, 1998.

[7] R. W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3):532–543, 1987.

[8] J. Liebowitz and J. Simien. Computational efficiencies for multi-agents: a look at a multi-agent system for sailor assignment. *Electonic government: an International Journal*, 2(4):384–402, 2005.

[9] D. Manlove. The structure of stable marriage with indifference. *Discrete Applied Mathematics*, 122(1-3):167–181, 2002.

[10] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Stability in matching problems with weighted preferences. In *Proc. ICAART'11*. SciTePress, 2011.

[11] A. E. Roth. The economics of matching: Stability and incentives. *Mathematics of Operations Research*, 7:617–628, 1982.

[12] A. E. Roth. Deferred acceptance algorithms: History, theory, practice, and open questions. *International Journal of Game Theory, Special Issue in Honor of David Gale on his 85th birthday*, 36:537–569, 2008.

[13] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. Gale-shapley stable marriage problem revisited: Strategic issues and applications. *Manage. Sci.*, 47(9):1252–1267, 2001.

Distance-Based Judgment Aggregation of Three-Valued Judgments with Weights

Marija Slavkovik and Wojciech Jamroga
Computer Science and Communication, University of Luxembourg

Abstract

Judgment aggregation theory studies how to amalgamate individual opinions on a set of logically related issues into a set of collective opinions. Aggregation rules proposed in the literature are sparse. All proposed rules consider only two-valued judgments, thus imposing the strong requirement that an agent cannot abstain from giving judgments on any of the issues. All proposed rules are also insensitive to weights that can be assigned to different judgments. We construct a family of weight-sensitive rules for aggregating individual judgment sets with abstentions. We do so by generalizing known distance-based judgment aggregation rules. We study the relations between existing distance-based rules and the rules we propose and the computational complexity of the winner determination problem.

1 Introduction

The theory of judgment aggregation studies the problem of aggregating individual answers to a set of binary interconnected questions, called an *agenda*. The answers, *i.e.*, judgments, given on some of the questions constrain the judgments that can consistently be given to others. Consequently, an agreement on the collective set of answers cannot always be reached by statistical pooling, one-by-one, the individual judgments [List and Polak, 2010].

Judgment aggregation is a relatively new field of social choice and it has been predominantly focused on studying the (im)possibility of rules for aggregation with respect to the fairness rules they can simultaneously satisfy. Few judgment aggregation rules have been constructed: the *premise-based* procedure, proposed in [Kornhauser and Sager, 1993] as "issue-by-issue voting" and studied in [Dietrich and Mongin, 2010; Mongin, 2008], *sequential procedures* [List, 2004; Dietrich and List, 2007; Li, 2010], and *distance-based merging procedures* [Pigozzi, 2006; Miller and Osherson, 2009; Endriss *et al.*, 2010]. All of these aggregation rules are defined for complete sets of judgments, i.e., the agents are not allowed to abstain from judgment. Furthermore, all the proposed rules satisfy the property of *anonymity*. The outcome of an anonymous aggregation rule depends only on the judgment sets being aggregated but not on the identity of the source or the nature of the agenda element. We argue that the proposed rules as such are insufficient to cover all judgment aggregation scenarios.

Consider a team that has to determine whether to purchase a new production robot.[1] The team makes the decision based on several factors such as: is the price affordable, is the robot production capacity adequate, is the robot easy to manipulate, etc. The team consists of a design engineer, a manager of the production unit that will use the robot, a purchasing agent, and a person who will be trained to operate the robot. The agents have different areas of expertise and each can address different domains of the purchasing problem. For instance, the design engineer can justifiably choose not to make a judgment on whether the robot is easy to manipulate, while the purchasing agent and the line manager may have different views about how important the price is, even if they have access to the same information.

In situations like this, not all team members need to give their judgments on all the agenda elements. The expertise of the agents may be distributed over the team members with no one member possessing all the relevant information. Furthermore, even when team members make judgments on the same agenda element, they may weigh their judgments differently. The aggregation of their judgments should account for abstentions, but also for different weights assigned to different judgments. The judgment aggregation rules proposed in the literature are not weight-sensitive and they are not designed to handle abstentions. The aim of this paper is to contribute towards filling this gap.

Frameworks of judgment aggregation in which agents are allowed to abstain from giving some judgments have been proposed in [Gärdenfors, 2006; Dokow and Holzman, 2010], but no aggregation rules were given. The challenge in aggregating three-valued judgments is in the decision on how to treat the case when an agent chooses to make no judgment. The abstentions can be interpreted along two dimensions, that of *semantics* and that of *relation* between abstentions and judgments. Abstaining can mean that the agent does not have enough information to make a judgment at present, that he thinks that a judgment cannot be made on that particular agenda element or maybe that he deems his opinion ir-

[1] This example is taken from [Ilgen *et al.*, 1991]

relevant. The chosen semantics of the abstention determines when a set of judgments that contains abstentions is consistent.

The second dimension of interpretation is the relation between an abstention regarding an agenda element and the judgments on that element. For instance, is the abstention an independent position in addition to "yes" and "no", or is it the half-way position between "yes" and "no"? The relation between abstentions and judgments determines the impact that abstentions have on which collective judgment is selected. There are several possibilities. Consider, for example, seven agents judging an issue p. Four of the agents abstain from making a judgment, two judge "yes" and one judges "no". On one hand, the collective judgment for p should be "yes" because this is the position of the majority of the agent's who do make a judgment. On the other hand, the majority of the agents abstain so the group should also abstain from giving a collective judgment on p.

In addition to rules that handle abstentions, we want to construct weight-sensitive rules. The only trivially weight-sensitive judgment aggregation rule considered in the literature is the *dictatorship rule*. Outside of judgment aggregation, weights associated with an agent have been considered in merging information by [Revesz, 1995], and we take the same approach. However, in addition to agent-associated weights, we also consider weights associated with a judgment, thus assigned to a $(judgment,\ agenda\ element)$ pair.

We develop our rules by generalizing the distance minimization approach to judgment aggregation since this approach is applicable to any agenda.[2] In contrast, the sequential aggregation rules are applicable only when there is a total order over the elements of the agenda, while the premise-based approach is applicable when the agenda can be partitioned to a set of *premises* and a set of *conclusions*. Moreover, as we show, the premise-based approach can be emulated by a distance-based aggregation rule.

This paper is structured as follows. In Section 2 we give the necessary preliminaries. In Section 3 the distance-based rules with abstentions are presented, and in Section 4 we propose the weight-sensitive version of these rules. In Section 5 we discuss the introduced rules and the computational complexity of the winner determination problem. In Section 6 we present our conclusions.

2 Preliminaries

There are two types of judgment aggregation frameworks: logic-based, [Dietrich, 2007], and abstract algebraic, [Rubinstein and Fishburn, 1986; Dokow and Holzman, 2010]. In a logic-based framework, the agenda is a set of formulas from a given logic. The agenda is closed under negation and a judgment set in this framework is a consistent subset of the agenda. In an abstract framework no agenda is given, instead, the agents choose from a set of allowed binary sequences. For example, if the agenda of the aggregation problem, in propositional logic, were $\{p, \neg p, p \rightarrow q, \neg(p \rightarrow q), q, \neg q\}$, then the corresponding set of allowed sequences in an abstract framework would be $\{\langle 0,1,0\rangle, \langle 0,1,1\rangle, \langle 1,0,0\rangle, \langle 1,1,1\rangle\}$. *E.g.,* $\{\neg p, p \rightarrow q, q\}$ is a judgment set for this agenda but $\{p, p \rightarrow q, \neg q\}$ is not.

Abstentions can be represented in several ways depending on the framework used. In a propositional logic framework, one can introduce a new agenda element $\overline{p}$ for each pair $\{p, \neg p\}$ to represent "the agent makes no judgment on p" while imposing the additional consistency constraints to denote that neither $\{\overline{p}, p\}$ nor $\{\overline{p}, \neg p\}$ are consistent sets. With this approach there is no need to extend the existing judgment aggregation rules and one can skip directly to constructing weight-sensitive rules. However, adding agenda elements in this way, as we show in Section 5, taxes the time it takes to compute the collective judgment set. [Dokow and Holzman, 2010] use a special symbol $*$, which is interpreted as a variable taking values from $\{0,1\}$, to represent abstentions in an abstract aggregation framework. This approach, as is the case with any abstract argumentation framework, requests for all of the allowed judgment sets to be explicitly given. The number of possible judgment sequences is exponential with respect to the cardinality of the sequences considered and taxes the space it takes to compute the collective judgment set.

We choose to use a ternary logic-based framework, in which the consistency of a judgment set is determined by a consequence relation. This allows us to keep the agenda as a set not closed under negation, and removes the need for all of the allowed judgment sets, or sequences, to be explicitly stated and stored.

2.1 Ternary logic framework

The choice of a three-valued logic determines the semantics of the abstention. In the ternary logic of Łukasiewicz, [Łukasiewicz, 1920; Urquhart, 2001], the third value is $\frac{1}{2}$, set in the middle of 0, *i.e.*, "false" and 1, *i.e.*, "true". This third value denotes "to be determined later". The Łukasiewicz semantics corresponds to the semantics of the symbol $*$ used by [Dokow and Holzman, 2010]. In the ternary logic of Kleene, [Kleene, 1938], the values that a formula can take are $\{T, I, F\}$, where the third value I denotes "undefined", for this logic also the numerical value set $\{0, \frac{1}{2}, 1\}$ is used with $I \equiv \frac{1}{2}$. In the context of judgment aggregation, the "to be determined later" means that when an agent is abstaining it is because he does not know the value of the agenda element at the moment of casting judgment, "undefined" means that the abstaining agent does not think that a judgment on the agenda element can be made. Other choices for ternary logics can also be made. For instance, the ternary logic of Bochvar interprets the third value as "meaningless" and any formula that has a meaningless component as meaningless, [Urquhart, 2001].

The choice of semantics can be based on the aggregation context in which the rule is used. For instance, the logic of Łukasiewicz is better suited to dynamic aggregation contexts in which agents give judgments to the same agenda several times, since the agents can make a judgment on p in the second round, even though they abstained in the firs. For the

[2]Note that aggregating multi-valued information by distance based merging has been already considered in the literature [Condotta *et al.*, 2008; Coste-Marquis *et al.*, 2007], but only outside of judgment aggregation.

same reason, the Kleene logic can be considered suited for aggregation problems in which the judgments are made only once.

We give a short overview of the logics of Kleene and Łukasiewicz. The syntax of the both of propositional logic (in BNF) $\mathcal{L}_{Prop}$:

$$\varphi ::= \top \mid \bot \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi,$$

where $p \in \mathcal{L}_0$ ranges over the set of atomic formulas. The formulas of $\mathcal{L}_{Prop}$ are assigned values from the set $T = \{0, \frac{1}{2}, 1\}$; $v(\top) = 1$ and $v(\bot) = 0$. The semantics of the non-atomic formulas according to Łukasiewicz is: $v(\neg\varphi) = 1 - v(\varphi)$; $v(\varphi_1 \wedge \varphi_2) = min(v(\varphi_1), v(\varphi_2))$; $v(\varphi_1 \vee \varphi_2) = max(v(\varphi_1), v(\varphi_2))$; $v(\varphi_1 \rightarrow \varphi_2) = min(1, 1 - v(\varphi_1) + v(\varphi_2))$ and $v(\varphi_1 \leftrightarrow \varphi_2) = 1 - |v(\varphi_1) - v(\varphi_2)|$.

The semantics according to Kleene is: $v(\neg\varphi) = 1 - v(\varphi)$; $v(\varphi_1 \wedge \varphi_2) = min(v(\varphi_1), v(\varphi_2))$; $v(\varphi_1 \vee \varphi_2) = max(v(\varphi_1), v(\varphi_2))$; with $\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$ and $\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$.

The consequence operator for a ternary logic, *e.g.*, $\models_L$ and $\models_K$, is defined in the standard way, [Urquhart, 2001]. Given a set of formulas $\Gamma \subset \mathcal{L}_{Prop}$ and a formula $\psi \in \mathcal{L}_{Prop}$, ψ is entailed by Γ, if for all assignments v, if $v(\psi) = 1$ for all formulas $\psi \in \Gamma$, then $v(\varphi) = 1$. A formula ψ for which $\emptyset \models_L \psi$ is a tautology of the Łukasiewicz logic, if $\emptyset \models_K \psi$ then ψ is a tautology of the Kleene logic. If $\Gamma \models_L \bot$, then Γ is inconsistent in the Łukasiewicz logic. If $\Gamma \models_K \bot$ then Γ inconsistent in the Kleene logic. *E.g.,* $p \rightarrow p$ is a tautology of the Łukasiewicz logic, but not of the Kleene logic.

The Łukasiewicz logic, together with $\models_L$, is a member of the set of general logics defined [Dietrich, 2007], thus for this logic all the impossibility results shown by [Dietrich, 2007] hold. The Kleeny logic is not a member of this set of general logics. We give the basic definitions using the Łukasiewicz logic framework. The definitions using any other ternary logic framework can be constructed in the same way.

2.2 Judgment aggregation definitions

A judgment aggregation problem is specified by a sequence of logically related issues called an agenda A. In our framework, the issues are well-formed formulas of $\mathcal{L}_{Prop}$. The logic relations between the agenda issues can also be given in addition to the agenda, by a set of formulas $\mathcal{R}$. For example, in the agenda $\mathcal{A} = \{a_1, a_1 \rightarrow a_2, a_2\}$ the elements are logically related, but in the well-known judgment aggregation problem, the "doctrinal paradox of [Kornhauser and Sager, 1993] where $\mathcal{A} = \{a_1, a_2, a_3\}$, the relations of the elements are specified by the additional set of formulas $\mathcal{R} = \{(a_1 \wedge a_2) \leftrightarrow a_3\}$ is given.

A judgment for issue $a \in \mathcal{A}$ is a valuation $v : \mathcal{L}_{Prop} \mapsto \{0, \frac{1}{2}, 1\}$. Note that by adopting this definition we consider an abstention as a judgment. Given a set of n agents N, the judgments rendered by an agent $i \in N$ on all m issues of $\mathcal{A}$ is called a *judgment sequence* $A_i \subseteq \{0, \frac{1}{2}, 1\}^m$. $A(a_j)$ is the judgment on element a_j according to sequence A. We can always create a judgment set A° from a judgment sequence A, and *vice versa*. We say that a judgment sequence A corresponds to a judgment set A° if and only if, for all issues $a \in \mathcal{A}$: $a \in A^\circ$ if and only if $v(a) = 1$; $\neg a \in A^\circ$ if and only if $v(a) = 0$; $a \notin A^\circ$ and $\neg a \notin A^\circ$ if and only if $v(a) = \frac{1}{2}$.

It is usually assumed, and we assume it here, that the judgment sets of all agents are consistent with respect to $\mathcal{R}$ *i.e.*, $\mathcal{A}_i^\circ \cup \mathcal{R} \not\models_L \bot$. The set of all judgment sets which are consistent with respect to $\mathcal{A}$ and $\mathcal{R}$ are denoted by $\Phi^\circ(\mathcal{A}, \mathcal{R}, \models_L)$; the set of its corresponding judgment sequences is denoted by $\Phi(\mathcal{A}, \mathcal{R}, \models_L)$. When $\mathcal{A}$, $\mathcal{R}$ and $\models_L$ are clear we write simply Φ and Φ°. Any subset of Φ which satisfies constraints X is denoted $\Phi^{\downarrow X}$. *E.g.,* the subset of Φ in which all sequences contain only judgments from $\{0, 1\}$ is denoted by $\Phi^{\downarrow\{0,1\}}$.

A *profile* is a $n \times m$ matrix $\pi = [p_{i,j}], p_{i,j} \in \{0, \frac{1}{2}, 1\}$ containing judgments of all agents $i \in N$ over all agenda issues $a \in \mathcal{A}$. If the profile consists only of consistent judgment sequences, then $\pi \in \Phi^n$. A line in the matrix, denoted π_i, corresponds to agent i's judgment sequence. A column in the matrix, denoted π^j, corresponds to the vector of all judgments rendered for $a_j \in \mathcal{A}$.

A judgment aggregation function, for a set of n agents is a function $f : \Phi^n \mapsto \Phi$. A judgment aggregation rule is a correspondence $F : \Phi^n \mapsto \mathcal{P}(\Phi)$, where $\mathcal{P}(\Phi)$ is the power set of Φ. A judgment sequence that is outputted from an aggregation rule is called a *collective judgment sequence.*

3 Distance-based judgment aggregation

A distance-based judgment aggregation procedure is, according to [Endriss *et al.*, 2010], a judgment aggregation rule $DBP : (\Phi^{\downarrow\{0,1\}})^n \mapsto \mathcal{P}(\Phi^{\downarrow\{0,1\}})$ defined as:
$DBP(\pi) = \arg\min_{A \in \Phi^{\downarrow\{0,1\}}} \sum_{i=1}^{n} \sum_{j=1}^{m} |(A(a_j) - p_{i,j}|$.
DBP can be generalized, in the style of the belief distance-based merging operators, see for example [Konieczny *et al.*, 2004], to a judgment aggregation rule $D^{d,\odot}$ by replacing the *aggregation function* $\sum$ with a general aggregation function $\odot$ and the Hamming *distance* by some distance d.

An aggregation function $\odot : (\mathcal{R}^+)^n \mapsto \mathcal{R}^+$ is any function that satisfies non-decreasingness, minimality and identity. The function $\odot$ is non-decreasing when, if $x \leq y$, then $\odot(x_1, \ldots, x, \ldots, x_n) \leq \odot(x_1, \ldots, y, \ldots, x_n)$. It satisfies minimality when $\odot(x_1, \ldots, x_n)$ has a unique absolute minimum $k \geq 0$ for $x_1 = \ldots = x_n = 0$ and identity when $\odot(x, \ldots, x) = x$. A distance $d : \{0, 1\}^m \times \{0, 1\}^m \mapsto \mathbb{R}^+$ is any total function which, for any $A, A' \in dom(d)$, satisfies: $d(A, A') = 0$ if and only if $A = A'$; $d(A, A') = d(A', A)$ and $d(A, A') + d(A', A'') \geq d(A, A'')$. The most common $\odot$ are $\sum$ and max, while the most common d are the Hamming distance, and the drastic distance d_D. The latter is defined as $d_D(A, A') = 0$ if and only if $A = A'$, and $d_D(A, A') = 1$ otherwise.

It is straightforward to extend the rule $D^{d,\odot}$ to aggregate three-valued judgment sequences.

Definition 1 *The three-valued distance-based judgment aggregation rule $\Delta^{d,\odot}$ is a rule $\Delta^{d,\odot} : \Phi^n \mapsto \mathcal{P}(\Phi)$ such that: $\Delta^{d,\odot}(\pi) = \arg\min_{A \in \Phi} \odot(d(A, \pi_1), \ldots, d(A, \pi_n))$. Where $\odot$ is as an aggregation function and d is a distance.*

Apart from the $\sum$ and the max, we can also use another well known aggregation function, the product $\prod$ [Grabisch *et al.*,

2009], with minor adjustments. We can define the function $\prod$ as $\prod : \prod_{i=1}^{n}(\epsilon + d(A, A_i))$, where $\epsilon \in \mathbb{R}^+$. We need to add the non-null constant ϵ to each distance to avoid multiplying with zero. Observe that $\prod(x_1, \ldots, x_n)$ has a unique absolute minimum in $k = \epsilon$ for $x_1 = \cdots = x_n = 0$.

We give some examples of distance. The drastic distance d_D can be used defined in the same way as for the case of two-valued judgments. The Hamming distance d_H can be defined as $d_H(A, A') = \sum_{i=1}^{m} \delta_h(A(a_i), A'(a_i))$ where $\delta_h(x_1, x_2) = 0$ iff $x_1 = x_2$; $\delta_h(x_1, x_2) = 1$ otherwise. We can use one more well-known distance metric, the *taxicab distance*[3] d_T. The d_T is defined as $d_T(A, A') = \sum_{i=1}^{m} |A(a_i) - A'(a_i)|$. As it can be observed, the d_T collapses into the d_H whenever both the judgment sequences compared are from $\Phi^{\downarrow\{0,1\}}$.

3.1 Basic judgment aggregation properties of $\Delta^{d,\odot}$

The basic properties considered for judgment aggregation are *universal domain*, *anonymity* and *independence of irrelevant alternatives*(IIA) [List and Polak, 2010]. Universal domain is satisfied when the domain of the aggregation rule includes Φ. (IIA) is satisfied when the collective judgment on any $a_j \in \mathcal{A}$ depends only on π^j. Anonymity is satisfied when the collective judgment set for a profile π is the same as the the collective judgment set of any permutation $\sigma(\pi)$.

The properties of universal domain, anonymity and independence of irrelevant alternatives can be extended to apply to aggregation rules as well. The rule $\Delta^{d,\odot}$ satisfies universal domain by construction. The independence of irrelevant alternatives does not hold for $\Delta^{d,\odot}$ and can be demonstrated by an example.

Whether $\Delta^{d,\odot}$ satisfies anonymity depends only on the selected aggregation function $\odot$ and not on the choice of distance. This is because all distances are by definition symmetric functions. $\Delta^{d,\odot}$ satisfies anonymity if and only if $\odot$ is symmetric. When π is a profile and $\hat{\pi} = \sigma(\pi)$ its permutation, observe that if $\hat{\pi} = \sigma(\pi)$ then $(d(\hat{A}, \hat{\pi}_1), \ldots, d(\hat{A}, \hat{\pi}_n))$ is a σ permutation of $(d(\hat{A}, \pi_1), \ldots, d(\hat{A}, \pi_n))$, because $d(\hat{A}, \pi_i) = d(\hat{A}, \hat{\pi}_j)$ when $\pi_i = \hat{\pi}_j$. Consequently $\Delta^{d,\odot}(\pi) = \Delta^{d,\odot}(\hat{\pi})$ if and only if $\odot(\mathbf{x}) = \odot(\sigma(\mathbf{x}))$, $\mathbf{x} \in (\mathbb{R}^+)^n$. An aggregation function is *symmetric* when for all permutations σ, $\odot(\mathbf{x}) = \odot(\sigma(\mathbf{x}))$ (pg.22, [Grabisch *et al.*, 2009]).

All the aggregation functions we considered: max, $\sum$ and $\prod$ are symmetric. Thus $\Delta^{d,\odot}$ is symmetric for all pairs of $\odot \in \{max, \sum, \prod\}$, $d \in \{d_D, d_H, d_T, m\}$.

3.2 Distances and judgment-abstention relations

The impact of the abstentions on the collective judgments is determined by the selection of the distance d. The distance determines the relation between a judgment sequence with abstentions and one without. By choosing the Hamming or the drastic distance, the abstentions are treated as a third option, an alternative to "yes" and "no". The Taxicab distance treats the abstention as a position half-way between "yes" and "no". All of these distances allow for the possibility of an abstention to be part of the collective judgment set. More "distance" functions can be defined for the abstention to have a different impact. For example, the function m assigns the distance zero from any judgment to the third-value judgment, thus ignoring the abstentions in the profile:
$m(A, A') = \sum_{i=1}^{m} \lfloor |A(a_i) - A'(a_i)| \rfloor$.
The function m is not a distance function, but it can be used to specify a distance-based aggregation rule.

[3]The Taxicab, also known as Manhattan, distance was introduced by Hermann Minkowski (1864-1909).

4 $\Delta^{d,\odot}$ with weights

To be able to specify weight sensitive aggregation rules, we need to introduce a new property for the distance functions, that of *granularity*.

Definition 2 *A distance d is granular, if it can be represented as $d(A, A') = \circledast_{i=1}^{m} \delta(A \nabla a_i, A' \nabla a_i)$, where $\circledast$ is a symmetric aggregation function with a unique minimum in $k = 0$.*

From the distances we considered, d_T and d_H are granular, while d_D is not.

A weight is a number $w_{i,j} \in \mathbb{R}^+$, $w_{i,j} \geq 1$, and it denotes the relevance of the judgment of agent i on $a_j \in \mathcal{A}$. The *weight matrix* $W = [w_{i,j}]_{n \times m}$ is an input to a weight-sensitive distance-based aggregation rules.

The weight can be specified by the agent who makes the judgment or by the agent who aggregates the judgments. Its meaning is determined by the aggregation context. In contexts such as the example for the robot purchase given in the introduction, the weight is specified by the agent who makes the judgment and it denotes the relevance the agent assigns to a particular reason, *i.e.*, issue. An agent can assign a weight to an agenda element to denote his confidence in his judgment.

Weights can be used to encode the reputation an agent has regarding particular agenda elements. In this case the weights are assigned by the agent who aggregates the judgments. We show how weights can be constructed from reputation. Assume that $r_{i,j} \in [0, 1]$ is the normalized reputation of agent i regarding $a_j \in \mathcal{A}$. To construct the weights is to set $w_{i,j} = 1 + r_{i,j}$, thus maintaining that $w_{i,j} \geq 1$. When the reputation of the agent is 0 his weight is 1.

Definition 3 *Let d^g be a granular distance and W a weight matrix. A three-valued distance-based judgment aggregation rule with weights $\Delta_W^{d^g,\odot}$ is a rule $\Delta_W^{d^g,\odot} : \Phi^n \times (\mathbb{R}^+)^{n \times m} \mapsto \mathcal{P}(\Phi)$ such that:*
$\Delta_W^{d^g,\odot}(\pi, W) = \arg\min_{A \in \Phi} \odot_{i=1}^{n} \circledast_{j=1}^{m} w_{i,j} \cdot \delta(A(a_j), p_{i,j})$.

Observe that when an agent has an "untarnished" reputation $r_{i,j} = 1$ for an issue, the weighted aggregation rule would still not treat their judgment as a "veto". To achieve "veto" of one agent on an issue, the weights of the remaining agents on that issue need to be set to zero.

Assuming that we have available only the weight associated to an agent, we can construct a $n \times 1$ *weight vector* $\mathbf{w} = [w_i]$, $w_i \geq 1$. A three-valued distance-based judgment aggregation rule with agent-weights $\Delta_w^{d^g,\odot}$ is then defined as $\Delta_w^{d,\odot}(\pi, \mathbf{w}) = \arg\min_{A \in \Phi} \odot_{i=1}^{n} w_i \cdot d(A, \pi_i)$.

When each agent is an expert on different issues, one may want to consider an agent's judgments only on issues in his

area of expertise. The weights can be used to encode *subjective agendas*, *i.e.*, individually designated agenda subset $\mathcal{A}_i$. The weights on an agent i are zero for all agenda issues $a_j \notin \mathcal{A}_i$.

5 Some more properties

We first consider the relations between the distance-based judgment aggregation rules we defined. Let F and F' be two judgment aggregation rules defined over domains $dom(F)$ and $dom(F')$ correspondingly. We say that a F is included in in F', denoted $F \subset F'$, if $dom(F) \cap dom(F') \neq \emptyset$ and for each $\pi \in dom(F) \cap dom(F')$, $F(\pi) \subseteq F'(\pi)$.

Proposition 1 *The following inclusion relations hold* $D^{d,\odot} \subset \Delta^{d,\odot} \subset \Delta_w^{d,\odot}$, $\Delta_w^{d^g,\odot} \subset \Delta_W^{d^g,\odot}$ *and* $\Delta^{d,\odot} \subset \Delta_W^{d,\odot}$.

Proof: $D^{d,\odot} \subset \Delta^{d,\odot}$ holds since $\Phi^{\downarrow\{0,1\}} \subset \Phi$; $\Delta^{d,\odot} \subset \Delta_w^{d,\odot}$ holds since we can use the unary vector $\mathbf{u} = (1, 1, \ldots, 1)$ to achieve $\Delta^{d,\odot}(\pi, \mathbf{u}) = \Delta_w^{d,\odot}(\pi, \mathbf{u})$. $\Delta_w^{d^g,\odot} \subset \Delta_W^{d^g,\odot}$, because we can always represent $\Delta_w^{d,\odot}$ through $\Delta_W^{d^g,\odot}$ by setting $w_{i,j} = v_i$ for all $a_j \in \mathcal{A}$. We can always represent the rule $\Delta^{d,\odot}$ as a $\Delta_W^{d^g,\odot}$ rule by setting $w_{i,j} = 1$. ■

Since $\Delta_W^{d^g,\odot}$ subsumes $\Delta_w^{d^g,\odot}$, $\Delta^{d^g,\odot}$ and $D^{d,\odot}$, we can use it to aggregate the profiles for sets of agents for which different types of weights are available.

5.1 Co-domain restrictions

The co-domain of $\Delta^{d^g,\odot}$, $\Delta_w^{d^g,\odot}$ and $\Delta_W^{d^g,\odot}$ corresponds to the set Φ° of all judgment sets A° for $\mathcal{A}$ for which $A^\circ \cup \mathcal{R}$ is consistent. Consequently, the selected judgment sequences may have the undecided judgments in them, and the sequence in which all judgments are undecided is also a possible outcome. This might be undesirable, and one may want to allow only for sequences from $\{0,1\}^m$ to be in the co-domain of the aggregation rule.

Ensuring that the aggregate satisfies certain constraints X, such as containing only judgments from $\{0,1\}$, can be accomplished by restricting co-domain of the rule. The co-domain restricted $\Delta_W^{d^g,\odot}$ can be defined as:

$$\Delta_W^{d^g,\odot}(\pi, W, X) = \arg\min_{A \in \Phi^{\downarrow X}} \odot_{i=1}^n \circledast_{j=1}^m \; w_{i,j} \cdot \delta(A(a_j), p_{i,j}).$$

5.2 The premise-based procedure emulated

Restricting the co-domain can be used to engineer certain properties for certain issues, such as for example *adherence to majority*. A judgment $v(a)$ on $a \in \mathcal{A}$ adheres to majority, with respect to a profile π, if the number n_i of agents in π^j, for which $p_{i,j} = v(a_j)$ is greater then the number of agents n_j for which $p_{i,j} \neq v(a_j)$; $v(a) = \frac{1}{2}$ when $n_i = n_j$.

As we know from the impossibility results of [Dietrich, 2007], a judgment set A in which the collective judgment for each issue $a \in \mathcal{A}$ corresponds to the majority judgment in π^j may be such that $A^\circ \cup \mathcal{R} \models_L \bot$. However, for some subset of agenda issues, majority-adherence can be consistently guaranteed. For example, the premise-based procedure guarantees majority-adherence to a subset of the agenda called *premises*. Given a profile π, and a subset of selected issues $b \in \mathcal{A}$, we can define $\Phi^{\downarrow X}$ to be the subset of Φ in which all judgment sequences A are such that which $A(b)$ is majority-adherent with respect to π.

5.3 General complexity result for distance-based judgment aggregation

The *judgment distance-based winner determination problem* for agenda $\mathcal{A}$, set of rules $\mathcal{R}$, and a distance-based rule $\Delta^{d,\odot}$, is defined as follows:

Definition 4 (WinDet for $\Delta^{d,\odot}$)
Input: *Profile* $\pi \in (\Phi(\mathcal{A}, \mathcal{R}, \models_L))^n$, *sequence* $A \in \Phi(\mathcal{A}, \mathcal{R}, \models_L)$.
Output: *true iff* $A \in \Delta^{d,\odot}(\pi)$.

Proposition 2 *If* $\odot$ *and* d *are computable in polynomial time then WinDet for* $\Delta^{d,\odot}$ *is in* Σ_2^P.

We prove the inclusion by showing an algorithm for WinDet.

Algorithm: $\underline{WinDet(\pi, A)}$

1. guess a valuation v for the atoms in $\mathcal{A}$;
2. if v is a model for A and not $ExistBetter(\pi, A)$ then $return(true)$ else $return(false)$;

Oracle: $\underline{ExistBetter(\pi, A)}$

1. guess $A' \in \{0, \frac{1}{2}, 1\}^m$;
2. guess a valuation v' for the atoms in $\mathcal{A}$;
3. if v' is a model for A' and $\odot(d(A', \pi_1), \ldots, d(A', \pi_n)) > \odot(d(A, \pi_1), \ldots, d(A, \pi_n))$ then $return(true)$ else $return(false)$;

Two observations are worth pointing out. In the weighted case, a weight matrix W is also a part of the input. If $\circledast$ and δ are computable in polynomial time wrt the size of π and W, then so is d, and the above result can be easily adapted. Moreover, if the number of possible scores for $\Delta_W^{d,\odot}$ is known in advance and bounded by a polynomial in n, m then computing WinDet for $\Delta_W^{d,\odot}$ is in Θ_2^P (where $\Theta_2^P = \mathbf{P}^{\mathbf{NP}[\log \mathbf{n}]}$ is the class of problems solvable by a polynomial-time deterministic Turing machine asking at most $\mathcal{O}(log\ n)$ adaptive queries to an **NP** oracle).[4] This can be demonstrated by the following variation of the algorithm. Let Val be the set of possible scores. Also, for an ordered set X, let $med(X)$ denote the median of X, X^+ denote the subset of X from $med(X)$ up, and X^- the part below $med(X)$.

Algorithm: $\underline{WinDet(\pi, A)}$

1. $Poss := Val$;
2. repeat
3. $\quad k := med(Poss)$;
4. $\quad$ if $Exist(\pi, Poss^-)$ then $Poss := Poss^-$ else $Poss := Poss^+$;
5. until $|Poss| = 1$;
6. if $\odot(d(A, \pi_1), \ldots, d(A, \pi_n)) = med(Poss)$ then $return(true)$ else $return(false)$;

Oracle: $\underline{Exist(\pi, Poss)}$

1. guess $A \in \{0, \frac{1}{2}, 1\}^m$ and a valuation v;

[4] We thank an anonymous reviewer for hinting the property and sketching the proof.

2. if v is a model for A and $\odot(d(A,\pi_1),\ldots,d(A,\pi_n)) \in Poss$ then $return(true)$ else $return(false)$;

Again, the algorithm and the result can be easily adapted for the weighted case of $\Delta_W^{d,\odot}$.

6 Conclusions and future work

The literature on judgment aggregation assumes that all agents have to give their judgments on all agenda elements, which seems an important limitation in many scenarios. Moreover, the agents' judgments on the same issue must bear the same weight. In this paper, we make the first step towards filling the gap. Our rules are based on distance minimization, i.e., a rule is specified by an aggregation function $\odot$ and a distance d. Unlike the weight-sensitive distance-based aggregation rules studied in the theory of belief merging by [Revesz, 1995], our weights can be assigned to each pair of $(judgment,\ agenda\ element)$ and not only to agents.

The semantics of abstention is determined by the choice of the propositional ternary logic. Which semantics to choose can be determined by the aggregation setting. The relation of the abstention from judgment to the crisp (yes/no) judgments is determined by the choice of distance d. Formally, we construct a dual judgment aggregation framework based on propositional ternary logic. The framework is dual since we can represent the input from the agents both as subsets of $\overline{\mathcal{A}} = \{\neg a | a \in \mathcal{A}\} \cup \mathcal{A}$, and as a sequence from $\{0, \frac{1}{2}, 1\}^m$. We demonstrate the expressive power of our rules by showing how the co-domain can be constrained to ensure collective judgment sequences with desirable properties.

The worst-case complexity for computing the winner determination problem turns out to be at most Σ_2^P in general, and at most Θ_2^P under reasonable conditions. Note that the specific complexity bounds may depend on the actual choice of d and $\odot$. For example, the WinDet problem for the drastic distance d_D can be solved in linear time with respect to the number of agents and issues. In the future we intend to study further the properties of different $\Delta_W^{d,\odot}$ rules with respect to the choice of $(d, \odot)$.

Acknowledgements. Wojciech Jamroga acknowledges the support of the FNR (National Research Fund) Luxembourg under project S-GAMES – C08/IS/03.

References

[Condotta *et al.*, 2008] J.F. Condotta, S. Kaci, and N. Schwind. A framework for merging qualitative constraints networks. In *FLAIRS Conference*, pages 586–591, 2008.

[Coste-Marquis *et al.*, 2007] S. Coste-Marquis, C. Devred, and S. Konieczny. On the merging of dung's argumentation systems. *Artificial Intelligence*, 171(10-15):730–753, 2007.

[Dietrich and List, 2007] F. Dietrich and C. List. Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics*, 4(19):391 – 424, 2007.

[Dietrich and Mongin, 2010] F. Dietrich and P. Mongin. The premiss-based approach to judgment aggregation. *Journal of Economic Theory*, 145(2):562 – 582, 2010.

[Dietrich, 2007] F. Dietrich. A generalised model of judgment aggregation. *Social Choice and Welfare*, 28(4):529–565, June 2007.

[Dokow and Holzman, 2010] E. Dokow and R. Holzman. Aggregation of binary evaluations with abstentions. *Journal of Economic Theory*, 145(2):544 – 561, 2010.

[Endriss *et al.*, 2010] U. Endriss, U. Grandi, and D. Porello. Complexity of winner determination and strategic manipulation in judgment aggregation. In *Proceedings of the 3rd International Workshop on Computational Social Choice (COMSOC-2010)*. University of Düsseldorf, September 2010.

[Gärdenfors, 2006] P. Gärdenfors. A representation theorem for voting with logical consequences. *Economics and Philosophy*, 22(2):181–190, July 2006.

[Grabisch *et al.*, 2009] M. Grabisch, J-L. Marichal, R. Mesiar, and E. Pap. *Aggregation Functions*. Cambridge University Press, 1st edition, July 2009.

[Ilgen *et al.*, 1991] D. R. Ilgen, D. A. Major, J. R. Hollenbeck, and D. J. Sego. Decision making in teams: Raising an individual decision making model to the team level. Technical Report ADA244699, MICHIGAN STATE UNIV EAST LANSING, December 1991.

[Kleene, 1938] S. C. Kleene. On notation for ordinal numbers. *The Journal of Symbolic Logic*, 3(4):150–155, 1938.

[Konieczny *et al.*, 2004] S. Konieczny, J. Lang, and P. Marquis. da^2 merging operators. *Artificial Intelligence Journal*, 157:45–79, 2004.

[Kornhauser and Sager, 1993] L.A. Kornhauser and L.G. Sager. The one and the many: Adjudication in collegial courts. *California Law Review*, 81:1–51, 1993.

[Li, 2010] N. Li. Decision paths in sequential non-binary judgment aggregation. Technical report, Universitat Autònoma de Barcelona, 2010.

[List and Polak, 2010] L. List and B. Polak. Introduction to judgment aggregation. *Journal of Economic Theory*, 145(2):441 – 466, 2010.

[List, 2004] C. List. A model of path-dependence in decisions over multiple propositions. *American Political Science Review*, 3(98):495 – 513, 2004.

[Łukasiewicz, 1920] J. Łukasiewicz. O logice trójwartościowej. *Ruch Filozoficzny*, 5:170–171, 1920.

[Miller and Osherson, 2009] M. Miller and D. Osherson. Methods for distance-based judgment aggregation. *Social Choice and Welfare*, 32(4):575 – 601, 2009.

[Mongin, 2008] P. Mongin. Factoring out the impossibility of logical aggregation. *Journal of Economic Theory*, 141(1):100–113, 2008.

[Papadimitriou, 1994] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pigozzi, 2006] G. Pigozzi. Belief merging and the discursive dilemma: an argument-based account to paradoxes of judgment aggregation. *Synthese*, 152(2):285–298, 2006.

[Revesz, 1995] Peter Z. Revesz. On the semantics of arbitration. *International Journal of Algebra and Computation*, 7:133–160, 1995.

[Rubinstein and Fishburn, 1986] A. Rubinstein and P. C. Fishburn. Algebraic aggregation theory. *Journal of Economic Theory*, 38(1):63–77, February 1986.

[Urquhart, 2001] A. Urquhart. Basic many-valued logic. In D.M. Gabbay and F. Guenthener, editors, *Handbook of Philosophical Logic (Second edition)*, volume 2, pages 249–295. Kluwer Academic Publishers, 2001.

Manipulating Single-Elimination Tournaments in the Braverman-Mossel Model

Isabelle Stanton and **Virginia Vassilevska Williams**
Computer Science Department
UC Berkeley
{isabelle,virgi}@eecs.berkeley.edu

Abstract

We study the power of a tournament organizer in manipulating the outcome of a single elimination tournament by fixing the initial seeding. It is not known whether the organizer can efficiently fix the outcome of the tournament even if the match outcomes are known in advance. We generalize a result from prior work by giving a new condition such that the organizer can efficiently find a tournament bracket for which the given player will win the tournament. We then use this result to show that for most tournament graphs generated by the Braverman-Mossel model, the tournament organizer can (very efficiently) make a large constant fraction of the players win, by manipulating the initial bracket. This holds for very low values of the error probability, i.e. the generated tournament graphs are almost transitive. Finally, we obtain a trade-off between the error probability and the number of players that can efficiently be made winners.

Introduction

The study of election manipulation is an integral part of social choice theory. Results such as the Gibbard-Satterthwaite theorem [Gibbard, 1973; Satterthwaite, 1975] show that all voting protocols that meet certain rationality criteria are manipulable. The seminal work of [Bartholdi *et al.*, 1989; 1992] proposes to judge the quality of voting systems using computational complexity: a protocol may be manipulable, but it may still be good if manipulation is computationally expensive. This idea is at the heart of computational social choice.

The particular type of election manipulation that we study in this paper is called *agenda control* and was introduced in [Bartholdi *et al.*, 1992]: there is an election organizer who has power over some part of the protocol, say the order in which candidates are considered. The organizer would like to exploit this power to fix the outcome of the election by making their favorite candidate win. [Bartholdi *et al.*, 1992] focused on plurality and Condorcet voting, agenda control by adding, deleting, or partitioning candidates or voters. We study the balanced binary cup voting rule, also called a *single-elimination* tournament: the number of candidates is a power of 2; at each stage the remaining candidates are paired up and their votes are compared; the losers are eliminated and the winners move on to the next round, until only one candidate remains. The power of the election organizer is to pick the pairing of the players in each round. We assume that the organizer knows all the votes in advance, i.e. for any two candidates, he knows which candidate is preferred.

Single-elimination is prevalent in sports tournaments such as Wimbledon or March Madness. In this setting, a tournament organizer has some information, say from prior matches or from betting experts, about the winner in any possible player match. The organizer is to come up with a *seeding* of the players through which they are distributed in the tournament bracket. The question is, can the tournament organizer abuse this power to determine the winner of the tournament?

There is significant prior work on this problem. [Lang *et al.*, 2007] showed that if the tournament organizer only has probabilistic information about each match, then the agenda control problem is NP-hard. [Vu *et al.*, 2009; 2010] showed that the problem is NP-hard even when the probabilities are in $\{0, 1, 1/2\}$ and that it is NP-hard to obtain a tournament bracket that approximates the maximum probability that a given player wins within any constant factor. [Vassilevska Williams, 2010] showed that the agenda control problem is NP-hard even when the information is deterministic but some match-ups are disallowed. [Vassilevska Williams, 2010] also gave conditions under which one can fix the outcome of the tournament when the organizer knows each match outcome in advance. It is still an open problem whether one can always determine in polynomial time whether the tournament outcome can be fixed in this deterministic setting.

The binary cup is a complete binary voting tree. Other related work has studied more general voting trees [Hazon *et al.*, 2008; Fischer *et al.*, 2008], and manipulation by the players themselves by throwing games to manipulate single-elimination tournaments [Russell and Walsh, 2009].

The match outcome information available to the tournament organizer can be represented as a weighted or unweighted tournament graph, a graph such that for every two nodes u, v exactly one of (u, v) or (v, u) is an edge. An edge (u, v) signifies that u beats v, and a weight p on an edge (u, v) means that u will beat v with probability p. With this repre-

sentation, the agenda control problem becomes a computational problem on tournament graphs. The tournament graph structure which comes from real world sports tournaments or from elections is not arbitrary. Although the graphs are not necessarily transitive, stronger players typically beat weaker ones. Some generative models have been proposed in order to study real-world tournaments. In this work, we study the Braverman-Mossel model [2008]. The basic idea is that there is an underlying total order of the players and the outcome of every match is probabilistic. There is some *global* probability $p << 1/2$ with which a weaker player beats a stronger player. This probability represents outside factors which do not depend on the players' abilities, such as weather or sickness.

[Vassilevska Williams, 2010] has shown that when $p \geq \Omega(\sqrt{\ln n/n})$, with high probability, the model generates a tournament graph where one can efficiently fix a single-elimination bracket for *any* given player. Two natural questions emerge. The first is can we still make almost all players win with a smaller noise value? The second is can we relax the Braverman-Mossel model to allow a different error probability for each pair of players? We address both questions.

Contributions We study whether one can compute a winning single-elimination bracket for a *king* player when the match outcomes are known in advance. A king is a player K such that for any other player a, either K beats a, or K beats some other player who beats a. We show that in order for a winning bracket to exist for a king, it is sufficient for the king to be among the top third of the players when sorted by the number of potential matches they can win. Before our work only much stricter conditions were known, e.g. that it is sufficient if the king beats half of the players. Our more general result allows us to obtain better results for the Braverman-Mossel model as well.

There are $\log n$ rounds in a single-elimination tournament over n players, so a necessary condition for a player to be a winner is that it can beat at least $\log n$ players. We consider a generalization of the Braverman-Mossel model in which the error probabilities $p(i,j)$ can vary but are all lower-bounded by a global parameter p. The expected outdegree of the weakest player i is $\sum_j p(i,j) \geq p(n-1)$, and it needs to be $\geq \log n$, so we focus on the case when p is $\Omega(\log n/n)$, as this is a necessary condition for all players to be winners.

Our results focus on this lower bound on the noise threshold. We improve previous results and show that when a tournament is generated with $\Omega(\log n/n)$ noise, we are able to fix the tournament for almost the top half of the players. We also show that there is a trade-off between the amount of noise and the number of players that can be made winners: as the level of noise increases, the tournament can be fixed for more and eventually all of the players. While this result does not answer the question of whether it is computationally difficult to fix a single-elimination tournament in general, it does show that for tournaments we might expect to see in practice, manipulation can be easy.

Notation
$N^{out}(a) = \{v \mid (a,v) \in E\}$
$N^{out}_X(a) = N^{out}(a) \cap X$
$N^{in}(a) = \{v \mid (v,a) \in E\}$, $N^{in}_X(a) = N^{in}(a) \cap X$
$out(a) = \lvert N^{out}(a)\rvert$, $out_X(a) = \lvert N^{out}_X(a)\rvert$
$in(a) = \lvert N^{in}(a)\rvert$, $in_X(a) = \lvert N^{in}_X(a)\rvert$
$\mathcal{H}^{in}(a) = \{v \mid v \in N^{in}(a), out(v) > out(a)\}$
$\mathcal{H}^{out}(a) = \{v \mid v \in N^{out}(a), out(v) > out(a)\}$
$\mathcal{H}(a) = \mathcal{H}^{in}(a) \cup \mathcal{H}^{out}(a)$
$E(X,Y) = \{(u,v) \mid (u,v) \in E, u \in X, v \in Y\}$

Table 1: A summary of the notation used in this paper.

Braverman-Mossel Model – Formal Definition

The premise of the Braverman-Mossel (BM) model is that there is an implicit ranking π of the players by intrinsic abilities so that $\pi(i) < \pi(j)$ means i has strictly better abilities than j. For clarity, we will call $\pi(i)$ i. When i and j play a match there may be outside influences so that even if $i < j$, j might beat i. The BM model allows that weaker players can beat stronger players, but only with probability $p < 1/2$. Here, p is a global parameter and if $i < j$, j beats i with probability $1 - p$. A random tournament graph generated in the BM model, a (*BM tournament*), is defined as: for every i, j with $i < j$, add edge (i,j) independently with probability $1 - p$ and otherwise add (j,i).

We give a generalization of the BM model, the GBM model, in which j beats i with probability $p(j,i)$, where $p \leq p(j,i) \leq 1/2$ for all i, j with $i < j$, *i.e.* the error probabilities can differ but are all lower-bounded by a global p. A random tournament graph generated in the GBM model (*GBM tournament*) is defined as: for every i, j with $i < j$, add edge (i,j) independently with probability $1 - p(j,i)$ and otherwise add (j,i).

Notation and Definitions Unless noted otherwise, all graphs in the paper are tournament graphs over n vertices, where n is a power of 2, and all single-elimination tournaments are balanced. In Table 1, we define the notation used in the rest of this paper. For the definitions, let $a \in V$ be any node, $X \subset V$ and $Y \subset V$ such that X and Y are disjoint. Given a player $\mathcal{A}$, unless otherwise stated, A denotes $N^{out}(\mathcal{A})$ and B denotes $N^{in}(\mathcal{A})$.

The outcome of a round-robin tournament has a natural graph representation as a tournament graph. The nodes of a tournament graph represent the players in a round-robin tournament, and a directed edge (a,b) represents a win of a over b.

We will use the concept of a *king* in a graph. Although the definition makes sense for any graph, it is particularly useful for tournaments, as the highest outdegree node is always a king.

Definition 1. *A* king *in $G = (V,E)$ is a node $\mathcal{A}$ such that for every other $x \in V$ either $(\mathcal{A}, x) \in E$ or there exists $y \in V$ such that $(\mathcal{A}, y), (y, x) \in E$.*

We also use the notion of a *superking*.

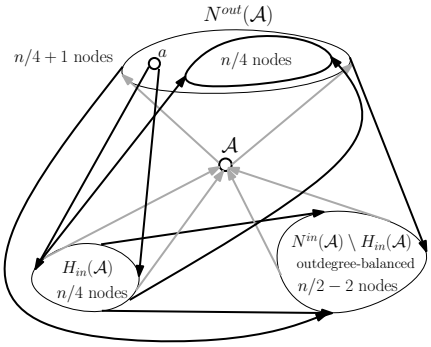


Figure 1: An example for which Theorem 1 does not apply, but for which Theorem 2 does apply.

Definition 2. *A* superking *in $G = (V, E)$ is a node $\mathcal{A}$ such that for every other $x \in V$ either $(\mathcal{A}, x) \in E$ or there exist $\log n$ nodes $y_1, \ldots, y_{\log n} \in V$ such that $(\mathcal{A}, y_i), (y_i, x) \in E$, $\forall i$.*

Kings that are also winners

A player being a king in the tournament graph is not a sufficient condition for it to also be able to win a single-elimination tournament. Consider that a player may be a king by beating only 1 player who, in turn, beats all the other players. [Vassilevska Williams, 2010] considered the question of how strong a king player needs to be in order for there to always exist a winning single-elimination tournament bracket for them.

Theorem 1. *[Vassilevska Williams, 2010] Let $G = (V, E)$ be a tournament graph and let $\mathcal{A} \in V$ be a king. One can efficiently construct a winning single-elimination tournament bracket for $\mathcal{A}$ if either*

$$\mathcal{H}^{in}(\mathcal{A}) = \emptyset, \text{ or } out(\mathcal{A}) \geq n/2.$$

We generalize the above result. The set $\mathcal{H}^{in}(\mathcal{A})$ represents all higher ranked nodes that beat the player $\mathcal{A}$. We show that it is sufficient for a player who is a king to only be as strong as the size of $\mathcal{H}^{in}(\mathcal{A})$.

Theorem 2 (Kings with High Outdegree). *Let G be a tournament graph on n nodes and $\mathcal{A}$ be a king. If $out(\mathcal{A}) \geq |\mathcal{H}^{in}(\mathcal{A})| + 1$, then one can efficiently compute a winning single-elimination bracket for $\mathcal{A}$.*

To see that the above theorem implies Theorem 1, note that if $out(\mathcal{A}) \geq n/2$, then $|\mathcal{H}^{in}(a)| \leq n/2 - 1 \leq out(\mathcal{A}) - 1$. Also, if $\mathcal{H}^{in}(\mathcal{A}) = \emptyset$ and $n \geq 2$, then $out(\mathcal{A}) \geq 1 \geq 1 + |\mathcal{H}^{in}(\mathcal{A})|$.

Theorem 2 is more general than Theorem 1. In Figure 1 we have an example of a tournament where Theorem 2 applies to the node $\mathcal{A}$ but not Theorem 1. Here, $|\mathcal{H}^{in}(\mathcal{A})| = \frac{n}{4}$, $|N^{out}(\mathcal{A})| = \frac{n}{4} + 1$ and the purpose of the node a is just to guarantee that $\mathcal{A}$ is a king. The example requires that each node in $N^{in}(\mathcal{A}) \setminus \mathcal{H}^{in}(\mathcal{A})$ have lower outdegree than $\mathcal{A}$ ($\frac{n}{4} + 1$) so we use an outdegree-balanced tournament for this set. This is a tournament where every vertex has outdegree equal to half the graph and it can be constructed inductively.

The intuition behind the proof of Theorem 2 is inspired by the results of [Stanton and Vassilevska Williams, 2011]. They show that a large fraction of highly ranked nodes can be tournament winners, provided a matching exists from the lower ranked to the higher ranked players. We are working with a king node so we are able to weaken the matching requirement. Instead, we carefully construct matchings that maintain that $\mathcal{A}$ is a king over the graph, while slowly eliminating the elements of $\mathcal{H}^{in}(\mathcal{A})$ until we reduce the problem to the case covered by Theorem 1.

We are now ready to prove Theorem 2. We will need a technical lemma from prior work relating the indegree and outdegree of two nodes. If a node $\mathcal{A}$ is a king then for every other node b, $N^{out}(\mathcal{A}) \cap N^{in}(b) \neq \emptyset$. This lemma is useful for showing a node is a king.

Lemma 1 ([Vassilevska Williams, 2010])**.** *Let a be a given node, $A = N^{out}(a), B = N^{in}(a)$, $b \in B$. Then $out(a) - out(b) = in_A(b) - out_B(b)$. In particular, $out(a) \geq out(b)$ if and only if $out_B(b) \leq in_A(b)$.*

Now we can prove Theorem 2.

Proof of Theorem 2: We will design the matching for each consecutive round r of the tournament. In the induced graph before the r^{th} round, let $\mathcal{H}_r$ be the subset of $\mathcal{H}^{in}(\mathcal{A})$ that is still live, A_r be the current outneighborhood of $\mathcal{A}$ and B_r be the current inneighborhood of $\mathcal{A}$. We will keep the invariant that if $B_r \setminus \mathcal{H}_r \neq \emptyset$, we have $|A_r| \geq |\mathcal{H}_r| + 1$, $\mathcal{A}$ is a king and the subset of nodes from the inneighborhood of $\mathcal{A}$ that have larger outdegree than $\mathcal{A}$ is contained in $\mathcal{H}_r$.

We now assume that the invariant is true for round $r-1$. We will show how to construct round r. If $\mathcal{H}_r = \emptyset$ we are done by reducing the problem to Theorem 1, so assume that $|\mathcal{H}_r| \geq 1$. We begin by taking a maximal matching M_r from A_r to $\mathcal{H}_r$. Since $|A_r| \geq |\mathcal{H}_r| + 1$, $A_r \setminus M_r \neq \emptyset$ i.e. M_r can not match all of A_r. Now, let M'_r be a maximal matching from $A_r \setminus M_r$ to $B_r \setminus \mathcal{H}_r$.

If $A_r \setminus (M'_r \cup M_r) \neq \emptyset$, there is some node a' leftover to match $\mathcal{A}$ to. Otherwise, pick any $a' \in M'_r \cap A_r$. Remove the edge matched to a' from M'_r and match a' with $\mathcal{A}$. To complete the matching, create maximal matchings within $\bar{A}_r = A_r \setminus (M'_r \cup M_r) \setminus \{a'\}$, $\bar{B}_r = B_r \setminus \mathcal{H}_r \setminus M'_r$ and $\mathcal{H}_r \setminus M_r$. Either zero or two of $|\bar{A}_r|, |\bar{B}_r|, |\mathcal{H}_r \setminus M_r|$ can be odd and so there are at most 2 unmatched nodes. These can be matched them against each other. Let M represent the union of all of these matchings.

We will now show that the invariants still hold. Notice that $\mathcal{A}$ is still a king on the sources of the created matching M. Now, consider any node b from $B_r \setminus \mathcal{H}_r$ which is a source in M. We have two choices. The first is that b survived by beating another node of B_r so it lost at least one outneighbor from B_r. Since M'_r was maximal, b may have lost at most one of its inneighbors (a'). Hence we still have

$$out_{B_{r+1}}(b) + 1 \leq (out_{B_r}(b) - 1 + 1) \leq$$
$$\leq in_{A_r}(b) - 1 \leq in_{A_{r+1}}(b).$$

By Lemma 1 this means that $out(b) \leq out(\mathcal{A})$. The second choice is if b survived by beating a leftover node $\bar{a}$ from A_r. This can only happen if $A_r \setminus (M'_r \cup M_r) \neq \emptyset$. Thus, $\bar{a}$ was in $A_r \setminus (M'_r \cup M_r)$. However, since M'_r was maximal, $\bar{a}$

must lose to b, and so all inneighbors of b from A_r move on to the next round, and again $out(b) \leq out(\mathcal{A})$. Hence $\mathcal{A}$ has outdegree at least as high as that of all nodes in $B_{r+1} \setminus \mathcal{H}_{r+1}$.

Now we consider A_{r+1} vs $\mathcal{H}_{r+1}$. We have

$$|A_{r+1}| \geq \lfloor (|A_r| + |M'_r| + |M_r| - 1)/2 \rfloor, \text{ and}$$

$$|\mathcal{H}_{r+1}| \leq \lceil (|\mathcal{H}_r| - |M_r|)/2 \rceil = \lfloor (|\mathcal{H}_r| + 1 - |M_r|)/2 \rfloor.$$

Since $|\mathcal{H}_r| \geq 1$ we must have $|M_r| \geq 1$. If either $|M_r| \geq 2$, $|A_r| \geq |\mathcal{H}_r|+2$, or $|M'_r| \geq 1$ then it must be that $|A_{r+1}| \geq \lfloor (|\mathcal{H}_r| + 2)/2 \rfloor \geq |\mathcal{H}_{r+1}| + 1$. Also, if $|\mathcal{H}_r|$ is even then

$$|A_{r+1}| \geq |\mathcal{H}_r|/2 = 1 + \lfloor (|\mathcal{H}_r| - 1)/2 \rfloor \geq |\mathcal{H}_{r+1}| + 1.$$

On the other hand, assume that $|M_r| = 1, |M'_r| = 0, |A_r| = |\mathcal{H}_r| + 1$ and $|\mathcal{H}_r|$ is odd. This necessarily implies that $|B_r \setminus \mathcal{H}_r| \leq 1$. Since $|A_r| = |\mathcal{H}_r| + 1$ is even, $|B_r|$ must be odd and so $|B_r \setminus \mathcal{H}_r|$ must be even. $|B_r \setminus \mathcal{H}_r|$ can only be 0. This means $|\mathcal{H}_r| = n_r/2 - 1$ (where n_r is the current number of nodes). We can conclude that $\mathcal{A}$ is a king with outdegree at least half the graph and the tournament can be efficiently fixed so that $\mathcal{A}$ wins by Theorem 1. □

Theorem 2 implies the following corollaries.

Corollary 1. *Let G be a tournament graph on n nodes and $\mathcal{A}$ be a king. If $|\mathcal{H}^{in}(\mathcal{A})| \leq (n-3)/4$, then one can efficiently compute a winning single-elimination tournament bracket for $\mathcal{A}$.*

Corollary 2. *Let G be a tournament graph on n nodes and $\mathcal{A}$ be a king in G. If $|\mathcal{H}(\mathcal{A})| \leq n/3 - 1$, then one can efficiently compute a winning single-elimination tournament bracket for $\mathcal{A}$.*

The proof of Corollary 1 follows by the fact that if $|\mathcal{H}^{in}(\mathcal{A})| = k$, then $out(\mathcal{A}) \geq (n-k)/3$. Corollary 2 simply states that any player in the top third of the bracket who is a king is also a tournament winner.

Proof of Corollary 2: Let $K = |\mathcal{H}(\mathcal{A})|$. Then the outdegree of $\mathcal{A}$ is at least $(n - K - 1)/2$. Let $h = |\mathcal{H}^{in}(\mathcal{A})|$. Then by Theorem 2, a sufficient condition for $\mathcal{A}$ to be able to win a single-elimination tournament is that $out(\mathcal{A}) \geq h+1$. Hence it is sufficient that $n-K-1 \geq 2h+2$, or that $2h+K \leq n-3$. Since $2h + K \leq 3K$, it is sufficient that $3K \leq n - 3$, and since $K \leq (n-3)/3$ we have our result. □

Braverman-Mossel Model

We can now apply our results to graphs generated by the Braverman-Mossel Model. From prior work we know that if $p \geq C\sqrt{\ln n/n}$ for $C > 4$, then with probability at least $1 - 1/\text{poly}(n)$, any node in a tournament graph generated by the BM model can win a single-elimination tournament. However, since p must be less than $1/2$, this result only applies for $n \geq 512$. Moreover, even for $n = 8192$ the relevant value of p is $> 13\%$ which is a very high noise rate. We consider how many players can be efficiently made winners when p is a slower growing function of n. We show that even when $p \geq C \ln n/n$ for a large enough constant C, a constant fraction of the top players in a BM tournament can be efficiently made winners.

Theorem 3 (BM Model Winners for Lower p). *For any given constant $C > 16$, there exists a constant n_C so that for all $n > n_C$ the following holds. Let $p \geq C \ln n/n$, and let G be a tournament graph generated by the BM model with error p. Then with probability at least $1 - 3/n^{C/8-2}$, any node v with $v \leq n/2 - 5C\sqrt{n \ln n}$ can win a single-elimination tournament.*

This result applies for $n \geq 256$ and also reduces the amount of noise needed. For example, if $C = 17$ then when $n = 8192$, it is only necessary that $p < 2\%$, as opposed to $> 13\%$. This is a significant improvement. The proof of Theorem 3 uses Theorem 2 and Chernoff-Hoeffding bounds.

Theorem 4 (Chernoff-Hoeffding). *Let $X_1, \ldots, X_n$ be random variables with $X = \sum_i X_i$, $E[X] = \mu$. Then for $0 \leq D < \mu$, $Pr[X \geq \mu + D] \leq exp(-D^2/(4\mu))$ and $Pr[X < \mu - D] \leq exp(-D^2/(2\mu))$.*

Proof of Theorem 3: Let C be given. Consider j. The expected of the number n_j of outneighbors of j in G is

$$E[n_j] = (1-p)(n-j)+(j-1)p = n(1-p)-p-j(1-2p).$$

This is exactly where we use the BM model. Our result is not directly applicable to the GBM model because this is only a lower bound on the expectation of n_j in that model. We will show that with high probability, all n_j are concentrated around their expectations and that all nodes $j \leq n/2$ are kings.

Showing that each n_j is concentrated around its' expectation is a standard application of the Chernoff bounds and a union bound. Therefore, $2/n^{C^2/4} < 1/n^C$ for $C > 16$ and $n > 2$. with probability at least $1 - 1/n^{C-1}$ for every j, $|E[n_j] - n_j| \leq C\sqrt{n \ln n}$.

We assume n is large enough so that $n >> \sqrt{n \ln n}$. We also assume that $p \leq 1/4$ so that $1 \geq (1 - 2p) \geq 1/2$. Now fix $j \leq n/2$. By the concentration result, this implies that

$$n_j \geq 3n/4 - 1 - j - C\sqrt{n \ln n} \geq$$

$$n/4 - 1 - C\sqrt{n \ln n} \geq \varepsilon n,$$

where $\varepsilon = 1/8$ works. The probability that j is a king is quite high: the probability that some node z has no inneighbor from $N^{out}(j)$ is at most

$$n(1-p)^{n_j} \leq n(1 - C \ln n/n)^{(n/(C \ln n)) \cdot C\varepsilon \ln n}$$

$$\leq 1/n^{\varepsilon C - 1}.$$

By a union bound, the probability that some node j is not a king is at most $1/n^{\varepsilon C-2}$. Therefore, we can conclude that the probability that all the n_j are concentrated around their expectations and all nodes $j \leq n/2$ are kings is at least $1 - (1/n^{C-1} + 1/n^{\varepsilon C-2})$.

We now need to upper bound $|\mathcal{H}^{in}(j)|$. We are interested in how many nodes with $i < j + 2C\sqrt{n \ln n}/(1-2p)$ appear in $N^{in}(j)$: if we have an upper bound on them, we can apply Theorem 2 to get a bound on j. First, consider how small $n_j - n_i$ can be for any i:

$$n_j - n_i \geq (i-j)(1-2p) - 2C\sqrt{n \ln n}.$$

So for $i \geq j + 2C\sqrt{n \ln n}/(1-2p)$, $n_j \geq n_i$ with high probability. The expected number of nodes $i < j$ that appear in $N^{in}(j)$ is $(1-p)(j-1)$. By the Chernoff bound, the probability that at least $(1-p)(j-1)+C\sqrt{j \ln n}$ of the $j-1$ nodes less than j are in $N^{in}(j)$ is $\leq exp(-C^2 j \ln n/4j) = n^{-C^2/4}$. Therefore, with probability at least $1-1/n^{C^2/4}$, the number of such i is at most $(1-p)(j-1) + C\sqrt{j \ln n}$. By a union bound, this holds for all j with probability at least $1 - 1/n^{C^2/4-1}$. Now, we can say with high probability that $|\mathcal{H}^{in}(j)|$ is at most

$$(1-p)(j-1) + C\sqrt{j \ln n} + 2C\sqrt{n \ln n}/(1-2p) \leq$$

$$\leq (1-p)(j-1) + 5C\sqrt{n \ln n}.$$

By Theorem 2, for there to be a winning bracket for j, it is sufficient that $\mathcal{H}^{in}(j) < n_j$ or that

$$(1-p)(j-1) + 5C\sqrt{n \ln n} <$$

$$n(1-p) - p - j(1-2p) - C\sqrt{n \ln n}$$

. This is equivalent to

$$j < \frac{n(1-p)}{(2-3p)} + \frac{(1-2p)}{(2-3p)} - 6C\frac{\sqrt{n \ln n}}{(2-3p)}.$$

It is sufficient if

$$j < n/2 + \frac{pn}{(2(2-3p))} + \frac{(1-2p)}{(2-3p)} - 24C\sqrt{n \ln n}/5,$$

and so for all $j \leq n/2 - 5C\sqrt{n \ln n}$, there is a winning bracket for j with probability at least

$$1 - (2/n^{C-1} + 1/n^{\varepsilon C-2}) \geq 1 - 3/n^{C/8-2}.$$

□

Improving the result for the GBM model through perfect matchings.

Next, we show that there is a trade-off between the constant in front of $\log n/n$ and the fraction of nodes that can win a single-elimination tournament. The proofs are based on the following result by [Erdős and Rényi, 1964]. Let $B(n,p)$ denote a random bipartite graph on n nodes in each partition such that every edge between the two partitions appears with probability p.

Theorem 5 ([Erdős and Rényi, 1964])**.** *Let c_n be any function of n, then consider $G = B(n,p)$ for $p = (\ln n + c_n)/n$. The probability that G contains a perfect matching is at least $1 - 2/e^{c_n}$.*

For the particular case $c_n = \Theta(\ln n)$, G contains a perfect matching with probability at least $1 - 1/\text{poly}(n)$.

Lemma 2. *Let $C \geq 64$ be a given constant. Let $n \geq 16$. Let G be a GBM tournament for $p = C \ln n/n$. Then with probability at least $1 - 2/n^{C/32-1}$, G is such that one can efficiently construct a winning single-elimination tournament bracket for the node ranked 1.*

Proof. We will call the top ranked node s. We will show that with high probability s has outdegree at least $n/4$ and that every node in $N^{in}(s)$ has at least $\log n$ inneighbors in $N^{out}(s)$. This makes s a superking, and by [Vassilevska Williams, 2010], s can win a single-elimination tournament.

The probability that s beats any node j is $> 1/2$, the expected outdegree of s is $> (n-1)/2$. By a Chernoff bound, the probability that s has outdegree $< n/4$ is at most $exp(-(n-1)/16) << 1/n^{C/32-1}$. Given that the outdegree of s is at least $n/4$, the expected number of inneighbors in $N^{out}(s)$ of any particular node y in $N^{in}(s)$ is at least $(n/4) \cdot (C \ln n/n) = (C/4) \ln n$.

We can show that each node in $N^{in}(s)$ has at least $\log n$ inneighbors from $N^{out}(s)$ by using a Chernoff bound and union bound. By a Chernoff bound, the probability that y has less than $(C/8) \ln n$ inneighbors from $N^{out}(s)$ is at most $exp(-(C/32) \ln n) = 1/n^{C/32}$. By a union bound, the probability that some $y \in N^{in}(s)$ has less than $(C/8) \ln n$ inneighbors from $N^{out}(s)$ is at most $1/n^{C/32-1}$. Therefore, s is a superking is with probability at least $1-2/n^{C/32-1}$ where $n \geq 16$, $n/4 \geq \log n$, $C > 64$, and $(C/8) \ln n \geq \log n$. □

Lemma 2 concerned itself only with the player who is ranked highest in intrinsic ability. The next theorem shows that as we increase the noise factor, we can fix the tournament for an increasingly large set of players. As the noise level increases, we can argue recursively that there exists a matching from $\frac{n}{2}+1 \ldots n$ to $1 \ldots \frac{n}{2}$, and from $\frac{3n}{4}+1 \ldots n$ to $\frac{n}{2}+1 \ldots \frac{3n}{4}$ and so forth. These matchings form each successive round of the tournament, eliminating all the stronger players.

Theorem 6. *Let $n \geq 16$, $i \geq 0$ be a constant and $p \geq 64 \cdot 2^i \ln n/n \in [0,1]$. With probability at least $1 - 1/\text{poly}(n)$, one can efficiently construct a winning single-elimination tournament bracket for any one of the top $1 + n(1-1/2^i)$ players in a GBM tournament.*

Proof. Let G be a GBM tournament for $p = C2^i \ln n/n$, $C \geq 64$. Let S be the set of all $n/2^{i-1}$ players j with $j > n(1-1/2^{i-1})$. Let s be a node with $1 + n(1-1/2^{i-1}) \leq s \leq 1 + n(1-1/2^i)$. The probability that s wins a single-elimination tournament on the subtournament of G induced by S is high: there is a set X of at least $n/2^i - 1$ nodes that are after s. By Lemma 2, s wins a single-elimination tournament on $X \cup \{s\}$ with high probability $1 - \frac{2}{(n/2^i)^{C/32-1}}$.

In addition, by Theorem 5, with probability at least $1 - \frac{2}{(n/2^i)^{C-1}}$, there is a perfect matching from $X \cup \{s\}$ to $S \setminus (X \cup \{s\})$. For every $1 \leq k \leq i-1$, consider

$$A_k = \{x \mid 1 + n(1-1/2^k) \leq x\}, \text{ and}$$

$$B_k = \{x \mid 1 + n(1-1/2^{k-1}) \leq x \leq n(1-1/2^k)\}.$$

Then $A_{k-1} = A_k \cup B_k$, $A_k \cap B_k = \emptyset$, and $|A_k| = |B_k| = n/2^k$. Hence $p \geq C \ln |A_k|/|A_k|$ for all $k \leq i-1$. By Theorem 5, the probability that there is no perfect matching from A_k to B_k for a particular k is at most $2/(n/2^k)^{C2^{i-k}-1}$. This value is maximized for $k = i$, and it is $2/(n/2^i)^{C-1}$. Thus by a union bound, with probability at least $1 - 2i/(n/2^i)^{C-1} =$

$1-1/\text{poly}(n)$, there is a perfect matching from A_k to B_k, for every k.

Thus, with probability at least $1 - 1/\text{poly}(n)$, s wins a single-elimination tournament in G with high probability, and the full bracket can be constructed by taking the unions of the perfect matchings from A_k to B_k and the bracket from S. □

For the BM model we can strengthen the bound from Theorem 3 by combining the arguments from Theorems 3 and 6.

Theorem 7. *There exists a constant n_0 such that for all $n > n_0$ the following holds. Let $i \geq 0$ be a constant, and $p = 64 \cdot 2^i \ln n/n \in [0,1]$. With probability at least $1-1/\text{poly}(n)$, one can efficiently construct a winning bracket for any one of the top $n(1-1/2^{i+1}) - (80/2^{i/2})\sqrt{n \ln n}$ players in a BM tournament.*

As an example, for $p = 256 \ln n/n$, Theorem 7 says that any of the top $7n/8 - 40\sqrt{n \ln n}$ players are winners while Theorem 6 only gives $3n/4 + 1$ for this setting of p in the GBM model.

Proof. As in Theorem 6, for every $1 \leq k \leq i$, consider $A_k = \{x \mid 1 + n(1 - 1/2^k) \leq x\}$, and $B_k = \{x \mid 1 + n(1 - 1/2^{k-1}) \leq x \leq n(1 - 1/2^k)\}$. Then $A_{k-1} = A_k \cup B_k$, $A_k \cap B_k = \emptyset$, and $|A_k| = |B_k| = n/2^k$. By the argument from Theorem 6, w.h.p. there is a perfect matching from A_k to B_k, for all k.

Consider A_i. By Theorem 3, with probability $1 - 1/\text{poly}(n/2^i) = 1 - 1/\text{poly}(n)$, we can efficiently fix the tournament for any of the first $n/2^{i+1} - 5 \cdot 16\sqrt{(n/2^i)\ln(n/2^i)}$ nodes in A_i. Combining the construction with the perfect matchings between A_k and B_k, we can efficiently construct a winning tournament bracket for any of the top

$$n - n/2^i + n/2^{i+1} - 80\sqrt{(n/2^i)\ln(n/2^i)} \geq$$

$$\geq n(1 - 1/2^{i+1}) - (80/2^{i/2})\sqrt{n \ln n} \text{ nodes.}$$ □

Conclusions

In this paper, we have shown a tight bound (up to a constant factor) on the noise needed to fix a single-elimination tournament for a large fraction of players when the match outcomes are generated by the BM model. As this model is believed to be a good model for real-world tournaments, this result shows that many tournaments in practice can be easily manipulated. In some sense, this sidesteps the question of whether it is NP-hard to fix a tournament in general by showing that it is easy on examples that we care about.

Acknowledgements

The authors are grateful for the detailed comments from the anonymous reviewers. The first author was supported by the National Defense Science and Engineering Graduate Fellowship, the National Science Foundation Graduate Fellowship and partially supported by the National Science Foundation Grant CCF-0830797. The second author was supported by the National Science Foundation under Grant #0937060 to the Computing Research Association for the CIFellows Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Computing Research Association.

References

[Bartholdi *et al.*, 1989] J. Bartholdi, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice Welfare*, 6(3):227–241, 1989.

[Bartholdi *et al.*, 1992] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election. *Mathematical and Computer Modeling*, pages 27–40, 1992.

[Braverman and Mossel, 2008] M. Braverman and E. Mossel. Noisy sorting without resampling. In *Symposium on Discrete Algorithms (SODA)*, pages 268–276, 2008.

[Erdős and Rényi, 1964] P. Erdős and A. Rényi. On random matrices. *Publications of the Mathematical Institute Hungarian Academy of Science*, 8:455–561, 1964.

[Fischer *et al.*, 2008] F. Fischer, A. D. Procaccia, and A. Samorodnitsky. On voting caterpillars:approximating maximum degree in a tournament by binary trees. In *2nd International Workshop on Computational Social Choice (COMSOC)*, 2008.

[Gibbard, 1973] A. Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41, 1973.

[Hazon *et al.*, 2008] N. Hazon, P.E. Dunne, S. Kraus, and M. Wooldridge. How to rig elections and competitions. In *2nd International Workshop on Computational Social Choice (COMSOC)*, 2008.

[Lang *et al.*, 2007] J. Lang, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Winner determination in sequential majority voting. In *The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[Russell and Walsh, 2009] T. Russell and T. Walsh. Manipulating tournaments in cup and round robin competitions. In *Algorithmic Decision Theory*, 2009.

[Satterthwaite, 1975] M. A. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10, 1975.

[Stanton and Vassilevska Williams, 2011] I. Stanton and V. Vassilevska Williams. Rigging tournament brackets for weaker players. *The Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[Vassilevska Williams, 2010] V. Vassilevska Williams. Fixing a tournament. In *AAAI*, 2010.

[Vu *et al.*, 2009] T. Vu, A. Altman, and Y. Shoham. On the complexity of schedule control problems for knockout tournaments. In *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2009.

[Vu *et al.*, 2010] T. Vu, N. Hazon, A. Altman, S. Kraus, Y. Shoham, and M. Wooldridge. On the complexity of schedule control problems for knock-out tournaments. *Submitted to JAIR*, 2010.

Venetian Elections and Lot-based Voting Rules

Toby Walsh
NICTA and UNSW
Sydney, Australia
toby.walsh@nicta.com.au

Lirong Xia
Department of Computer Science
Duke University
Durham, NC 27708, USA
lxia@cs.duke.edu

Abstract

Between 1268 and 1797, the Venetian Republic used a complicated voting system that appears designed to resist manipulation. The system starts with randomly drawing voters, followed by 8 rounds of a complicated addition and elimination of voters before the approval voting rule is finally used to select the winner, the new *Doge*. In this paper, we study a family of voting rules inspired by this Venetian election system, which we call *lot-based voting rules*. Such rules have two steps: in the first step, k votes are selected by a lottery, then in the second round (the runoff), a voting rule is applied to select the winner based on these k votes. We study some normative properties of such lot-based rules. We also investigate the computational complexity of computing the winner with weighted and unweighted votes, and of computing manipulations. Finally, we propose an efficient sampling technique for generating the k runoff voters non-uniformly.

1 Introduction

A central question in computational social choice is whether computational complexity can protect elections from manipulation. For certain voting rules it is NP-hard for a potential manipulator to compute a beneficial manipulation.Modifications have even been proposed to tweak common voting rules to make manipulation NP-hard. [5; 10]. Such results need to be treated with caution since NP-hardness is only a worst-case notion and "hard" instances may be rare.See [11; 12] for recent surveys. Of course, if it is already computationally hard for a manipulator to compute the winner, then intuitively it is likely to be computationally hard for her to find a beneficial manipulation. In fact, computing the winner is NP-hard for Kemeny's, Dodgson's and Slater's rule [3; 1; 2; 6].

Surprisingly, the idea of intentionally using complexity to prevent manipulation of a voting system goes back at least seven centuries ago. Lines argues that "*The most enduring and perhaps the most complex electoral process is quite likely that used by the Venetian oligarchy to elect their dogi*" [14]. This multi-stage voting procedure was used between 1268 and the end of the Venetian Republic in 1797. The procedure consists of 10 rounds, with all but the last round constructing an electoral college for the next round, and the last round actually electing the Doge, the highest official in Venice. This procedure appears designed to resist manipulation, or at least to offer the appearance of doing so. Wolfson argues that " *The main idea . . . seems to have been to introduce a system of election so complicated that all possibility of corruption should be eliminated*" [18]. On the other hand, Mowbray and Gollmann suggest that it is "*security theatre*", containing "*actions which do not increase security, but which are designed to make the public think that the organization carrying out the actions is taking security seriously*" [15]. Nevertheless they also remark that it "*offers some resistance to corruption of voters*".

Our contributions. Venetian elections have two interesting features in all but the last round: (1) voters are eliminated randomly, and (2) the voters in the current round vote on the voters who go forwards to the next round. In this paper, we report some preliminary results on a family of voting rules inspired by the first feature of such Venetian elections, which we call *lot-based rules*. It would be interesting nevertheless to consider the second feature. Lot-based rules are composed of two steps: in the first step, k votes are selected by a lottery, then in the second step (the runoff), a voting rule (called the *runoff rule*) is applied to select the winner based on these k votes. We study some normative properties of the lot-based rules. We investigate the computational complexity of computing the winner of lot-based rules with weighted and unweighted votes, respectively, and of computing a manipulation. Finally, we propose an efficient sampling technique for generating the k runoff voters from non-uniform distributions. Our results suggest it will be interesting to study further the computational properties of such rules.

For lot-based rules, it is easy for the chair to compute the winner provided computing the winner for the runoff rule is easy. This is essentially different from Kemeny's rule, where computing the winner of a given profile is hard. On the other hand, in order for a manipulator to compute a beneficial false vote, she needs to compute the probability for a given candidate to win, which we will show to be computationally hard. The winner evaluation/computation problem we focus on in this paper is from the perspective of a manipulator.

Related work. Lot-based rules are a type of randomized voting rules. Gibbard [13] proved that when there are at least 3 candidates, if a randomized voting rule satisfies *Pareto opti-*

mality and a probabilistic version of strategy-proofness, then it must be a probability mixture of dictatorships (called *random dictatorships*). We note that any random dictatorship is a lot-based rule, where $k = 1$, and the runoff rule selects the top-ranked candidate as the winner when there is a single vote.

Conitzer and Sandholm [5] and Elkind and Lipmaa [10] studied another type of hybrid voting systems where manipulations are hard to compute. Their systems are composed of two steps: in the first step, a (possibly randomized) voting rule is used to rule out some candidates, and in the second step another voting rule (not necessarily the same as the one used in the first step) is used to select the winner from the remaining candidates. We note that in the first step of their systems, some *candidates* are eliminated, while in the first step of our lot-based rules, some *voters* are eliminated. In that sense, lot-based rules can also be seen as a universal tweak that adds a pre-round that randomly eliminates some voters, to make voting rules hard to manipulate. It would therefore be interesting to consider even more complex voting systems which do both.

2 Preliminaries

Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be the set of *candidates* (or *alternatives*). A linear order $\succ$ on $\mathcal{C}$ is a transitive, antisymmetric, and total relation on $\mathcal{C}$. The set of all linear orders on $\mathcal{C}$ is denoted by $L(\mathcal{C})$. An n-voter profile P on $\mathcal{C}$ consists of n linear orders on $\mathcal{C}$. That is, $P = (V_1, \ldots, V_n)$, where for every $j \leq n$, $V_j \in L(\mathcal{C})$. The set of all n-profiles is denoted by $\mathcal{F}_n$. We let m denote the number of candidates. A (deterministic) *voting rule* r is a function that maps any profile on $\mathcal{C}$ to a unique winning candidate, that is, $r : \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \to \mathcal{C}$. A *randomized voting rule* is a function that maps any profile on $\mathcal{C}$ to a distribution over $\mathcal{C}$, that is, $r : \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \to \Omega(\mathcal{C})$, where $\Omega(\mathcal{C})$ denotes the set of all probability distributions over $\mathcal{C}$. The following are some common voting rules. If not mentioned specifically, ties are broken in the fixed order $c_1 \succ c_2 \succ \cdots \succ c_m$.

• *(Positional) scoring rules*: Given a *scoring vector* $\vec{s}_m = (\vec{s}_m(1), \ldots, \vec{s}_m(m))$ of m integers, for any vote $V \in L(\mathcal{C})$ and any $c \in \mathcal{C}$, let $\vec{s}_m(V, c) = \vec{s}_m(j)$, where j is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $\vec{s}_m(P, c) = \sum_{j=1}^{n} \vec{s}_m(V_j, c)$. The rule will select $c \in \mathcal{C}$ so that $\vec{s}_m(P, c)$ is maximized. We assume scores are integers and decreasing. Example of positional scoring rules are *majority*, for which $m = 2$ and the scoring vector is $(1, 0)$; *Borda*, for which the scoring vector is $(m - 1, m - 2, \ldots, 0)$.

• *Approval*: Each voter submits a set of candidates (that is, the candidates that are "approved" by the voter). The winner is the candidate approved by the largest number of voters. Every voter can approve any number of candidates.

• *Voting trees*: A voting tree is a binary tree with m leaves, where each leaf is associated with an candidate. In each round, there is a pairwise election between an candidate c_i and its sibling c_j: if the majority of voters prefer c_i to c_j, then c_j is eliminated, and c_i is associated with the parent of these two nodes. The candidate that is associated with the root of the tree (i.e. wins all its rounds) is the winner. The rule that uses a balanced voting tree is also known as *cup*.

3 Electing the Doge

The electorate (which consisted of around the 1000 or so male members of the Maggior Consiglio aged 30 or over) were first reduced by a lottery to an electoral college of 30 voters. This college was then reduced again by a lottery to 9 voters.[1] These 9 then elected a college of 40 voters chosen from any of the electorate, all of whom had to receive 7 out of 9 approval votes.[2] These 40 were then reduced by a lottery to an electoral college of 12 voters. These 12 then elected a college of 25 voters, all of whom had to receive 9 out of 12 approval votes. These 25 were then reduced by a lottery to an electoral college of 9 voters. These 9 then elected a college of 45 voters, all of whom had to receive 9 out of 12 approval votes. These 45 were then reduced by a lottery to an electoral college of 11 voters. These 11 then elected a college of 41 voters, all of whom had to receive 9 out of 11 approval votes. In the tenth and final round, the electoral college of 41 voters elected the Doge, who was required to receive 25 or more approval votes from the 41 voters.

This itself is still a simplified description of the process. For example, the process of enlarging the electoral college by vote was itself complicated. Consider the third round of the election where the electoral college is enlarged from 9 members to 40. The first 4 of the 9 college members selected by the lottery in the second round each nominated 5 people (who each had to receive 7 out of 9 approval votes) whilst the last 5 of the 9 college members selected by the lottery in the second round each nominated 4 people (who also each had to receive 7 out of 9 approval votes). This gives a total of 40 nominated members in the electoral college for the fourth round. Similarly, in the fifth, seventh and ninth rounds when the electoral college was enlarged, each member of the college nominated in turn a small number of new members. As a second example of the additional complexity, only one person from each family was allowed to be selected by a lottery. All relatives of a person selected by a lottery were removed from the rest of that round. As a third example, none of the members of the electoral colleges of size 9, 11 or 12 were allowed to be members of the final electoral college of size 41. As a fourth example of the additional complexity, the vote in the final round was not a simple approval vote. In addition to their approval votes, each member of this final electoral college also nominated one candidate. These nominated candidates were considered in a random order, and the first candidate who was secured 25 approval votes was elected the Doge.

The voting procedure also changed in several ways over the centuries. For example, the penultimate round originally had an electoral college of 40 voters. However, after a tied vote in 1229, this was increased to 41 to reduce the chance of a tie. As a second example, as explained earlier, the final vote was originally sequential. However, at some later point, voting moved to simultaneous voting.

[1] It has been suggested that two rounds (instead of one) are used to reduce the electoral college of 9 votes largely for procedural ease. That is, it was difficult to reduce the size of election college to 9 with a single lottery.

[2] It was not specified what happens if none of the voters receive 7 approval votes.

4 Lot-based voting rules

Elections involving lotteries are not restricted to Venice. Many other Italian cities have used such elections as well. Lotteries were also used in the election of the Archbishop of Novgorod, one of the oldest offices in the Russian Orthodox Church. Indeed the use of lotteries in elections can be traced back to at least before the birth of Christ with elections in the city-state of Athens. One of the arguments advanced for using lotteries is their fairness and resistance to manipulation [8].

We consider therefore a family of lot-based voting rules that are guaranteed always to elect a winner. These rules are closely related to the procedure used to elect the Doge.

Definition 1. *Let X denote a voting rule (deterministic or randomized). We define a randomized voting rule LotThen$X$ as follows. Let k be a fixed number that is smaller than the number of voters. The winner is selected in two steps: in the first step, k voters are selected uniformly at random, then, in the second step, the winner is chosen by the voting rule X from the votes of the k voters selected in the first step.*

For instance, LotThenApproval is an instance of this rule in which the set of voters is first reduced by a lottery, and then a winner is chosen by approval voting. Lot-based rules are in practical use. For example, the Chair of the Internet Engineering Task Force is selected by a randomly chosen nominating committee of 10 persons who vote (using an unspecified rule) for the new Chair.

We emphasize that in the first step of lot-based rules, some *voters* are eliminated, while in the first step of voting systems studied by Conitzer and Sandholm [5] and Elkind and Lipmaa [10], some *candidates* are eliminated.

We first consider the axiomatic properties possessed by lot-based voting rules. As the rules are non-deterministic, we need probabilistic versions of the usual axiomatic properties defined as follows.[3]

Definition 2. A randomized voting rule r satisfies

• *anonymity*, if for any profile $P = (V_1, \ldots, V_n)$, any permutation π over $\{1, \ldots, n\}$, and any candidate c, we have $r(P)(c) = r(V_{\pi(1)}, \ldots, V_{\pi(n)})(c)$, where $r(P)(c)$ is the probability of c in the distribution $r(P)$.

• *neutrality*, if for any profile P, any permutation M over $\mathcal{C}$, and any candidates c, we have $r(P)(c) = r(M(P))(M(c))$.

• *unanimity*, if for any profile P where all voters rank c in their top positions, we have $r(P)(c) = 1$.

• *weak monotonicity*, if for any candidate c and any pair of profiles P and P', where P' is obtained from P by raising c in some votes without changing the orders of the other candidates, we have $r(P)(c) \leq r(P')(c)$.

• *strong monotonicity*, if for any candidate c and any pair of profiles $P = (V_1, \ldots, V_n)$ and $P' = (V'_1, \ldots, V'_n)$, such that for every $j \leq n$ and every $d \in \mathcal{C}$, $c \succ_{V_j} d \Rightarrow c \succ_{V'_j} d$, we have $r(P)(c) \leq r(P')(c)$.

• *Condorcet consistency*, if whenever there exists a candidate who beats all the other candidates in their pairwise elections, this candidate wins the election with probability 1.

[3]The definitions for the axiomatic properties for approval are omitted due to the space constraints.

When the voting rule is deterministic (i.e. the unique winner wins with probability 1), all these axioms reduce to their counterparts for deterministic rules. The next theorem shows that LotThenX preserves some of these axioms from X.

Theorem 1. *If the voting rule X satisfies anonymity/ neutrality/ (strong or weak) monotonicity/ unanimity, then for every k, LotThenX also satisfies anonymity/ neutrality/ (strong or weak) monotonicity/ unanimity.*

The proof is quite straightforward, and therefore is omitted due to space constraints. However, there are other properties that can be lost like, for instance, Condorcet consistency.

Theorem 2. *LotThenX may not be Condorcet consistent even when X is.*

Proof: Suppose $n = 2k+1$, $k+1$ voters vote in one order and the remaining k voters vote in the reverse order. The lottery may select only the votes of the minority, which means that the Condorcet winner does not win with probability 1. □

We note that when $n = k$, LotThenX becomes exactly X. Therefore, if X does not satisfy an axiomatic property, neither does LotThenX.

Theorem 3. *If LotThenX satisfies an axiomatic property for every k, then X also satisfies the same axiomatic property.*

5 Computing the winner

Lot-based voting rules are non-deterministic. Hence, even if we know all the votes, we can only give a probability that a certain candidate wins. Following [7], given a probability p in $[0, 1]$, we define EVALUATION as the decision problem of deciding whether a given candidate can win with a probability strictly larger than p. In this section, we show that lot-based voting rules provide some resistance to strategic behavior by making it computationally hard even to evaluate who may have won. In particular, we show that there exist deterministic voting rules for which computing the winner is in P, but EVALUATION of the corresponding lot-based voting rule is NP-hard. As is common in computational social choice, we consider both weighted voted with a small number of candidates, and unweighted votes with an unbounded number of candidates. Of course even if EVALUATION is hard, the manipulator may still be able to compute her optimal strategy in polynomial time. This issue will be discussed in Section 6.

5.1 Weighted votes

Theorem 4. EVALUATION *for LotThenCup is* NP*-hard when votes are weighted and there are three or more candidates.*

Proof: We give a reduction from a special SUBSET-SUM problem. In such a SUBSET-SUM problem, we are given $2k'$ integers $\mathcal{S} = \{w_1, \ldots, w_{2k'}\}$ and another integer W. We are asked whether there exists $S \subset \mathcal{S}$ such that $|S| = k'$ and the integers in S sum up to W. We consider the cup rule (balanced voting tree) where ties are broken in lexicographical order. We only show the proof for three candidates; other cases can be proved similarly. For any SUBSET-SUM instance, we construct an EVALUATION for LotThenCup instance as follows.

Candidates: $\mathcal{C} = \{a, b, c\}$. The cup rule has a play b and the winner of this play c. Let $k = k' + 1$.

Profile: For each $i \leq 2k'$, we have a vote $c \succ a \succ b$ of weight w_i. In addition, we have one vote $b \succ a \succ c$ of weight W. We consider the problem of evaluating whether candidate a can win with some probability strictly greater than zero.

If the lottery does not pick any $b \succ a \succ c$, then c wins for sure. If the lottery picks the vote $b \succ a \succ c$, then there are three cases to consider. In the first case, the sum of the weights of the other k' votes is strictly less than W. Then, b beats a in the first round, so a does not win. In the second case, the sum of the weights of the other k' votes is strictly more than W. Then, a beats b in the first round, but then loses to c in the second round, so a does not win. In the third case, the sum of weights of the other k' votes is exactly W. Then, a wins both rounds due to tie-breaking. Hence a wins if and only the sum of the weights of the remaining k' votes is exactly W. Thus the probability that a wins is greater than zero if and only if there is a subset of k' integers with sum W. □

Theorem 5. *There is a polynomial-time Turing reduction from* SUBSET-SUM *to* EVALUATION *for LotThenApproval with weighted votes and two candidates*[4] .

Proof sketch: Given any SUBSET-SUM instance $\{w_1, \ldots, w_{2k'}\}$ and W, we construct the following two types of EVALUATION for LotThenApproval instances: the profiles in both of them are the same, but the tie-breaking mechanisms are different. For each $i \leq 2k'$, there is a voter with weight w_i who approves candidate a. In addition, there is voter with weight W who approves b. Let P denote the profile and $k = k' + 1$. For any $p \in [0, 1]$, we let $A(p)$ (respectively, $B(p)$) denote the EVALUATION instance where ties are broken in favor of a (respectively, b), and we are asked whether the probability that a (respectively, b) wins for P is strictly larger than p. Then, we use binary search to search for an integer i such that $i \in [0, \binom{2k'+1}{k'} - \binom{2k'}{k'+1}]$ and the answers to both $A\left(1 - \frac{i+1}{\binom{2k'+1}{k'+1}}\right)$ and $B\left(\frac{i}{\binom{2k'+1}{k'+1}}\right)$ are "yes". If such an i can be found, then the SUBSET-SUM instance is a "yes" instance; otherwise it is a "no" instance. □

It follows that if EVALUATION for LotThenApproval with weighted votes and two candidates is in P, then P=NP.

5.2 Unweighted votes

Theorem 6. *With unweighted votes and an unbounded number of candidates,* EVALUATION *for LotThenBorda is* NP*-hard.*

Proof: We prove the NP-hardness by a reduction from the EXACT 3-COVER (X3C) problem. In an X3C instance, we are given a set $\mathcal{V} = \{v_1, \ldots, v_{3q}\}$ of $3q$ elements and $\mathcal{S} = \{S_1, \ldots, S_t\}$ such that for every $i \leq t$, $S_i \subseteq \mathcal{V}$ and $|S_i| = 3$. We are asked whether there exists a subset $J \subseteq \{1, \ldots, t\}$ such that $|J| = q$ and $\bigcup_{j \in J} S_j = \mathcal{V}$.

For any X3C instance $\mathcal{V} = \{v_1, \ldots, v_{3q}\}$ and $\mathcal{S} = \{S_1, \ldots, S_t\}$, we construct an EVALUATION instance for LotThenBorda as follows.

Candidates: $\mathcal{C} = \{c\} \cup \mathcal{V} \cup D$, where $D = \{d_1, \ldots, d_{3q^2}\}$. Let $k = q$ and $p = 0$.

[4]The proof can be easily extended to any LotThenX where X is the same as the majority rule when there are only two candidates.

Profile: For each $j \leq t$, we let $V_j = [(\mathcal{S} \setminus S_j) \succ c \succ D \succ S_j]$. The profile is $P = (V_1, \ldots, V_t)$.

Suppose the EVALUATION instance has a solution. Then, there exists a sub-profile P' of P such that $|P'| = q$ and Borda$(P') = c$. Let $P' = (V_{i_1}, \ldots, V_{i_q})$. We claim that $J = \{i_1, \ldots, i_q\}$ constitutes a solution to the X3C instance. Suppose there exists a candidate $v \in \mathcal{V}$ that is not covered by any S_j where $j \in J$. Then, v is ranked above c in each vote in P', which contradicts the assumption that c is the Borda winner.

Conversely, let $J = \{i_1, \ldots, i_q\}$ be a solution to the X3C instance. Let $P' = (V_{i_1}, \ldots, V_{i_q})$. It follows that for each $v \in \mathcal{V}$, the Borda score of c minus the Borda score of v is at least $3q^2 - (3q-3) \times q > 0$. For each $d \in D$, c is ranked above d in each vote in P'. Therefore, c is the Borda winner, which means that the EVALUATION instance is an "yes" instance. □

Theorem 7. *With unweighted votes and an unbounded number of candidates, computing the probability for a given candidate to win under LotThenBorda is #*P*-complete.*

Proof: We prove the theorem by a reduction from the #PERFECT-MATCHING problem. Given three sets $X = \{x_1, \ldots, x_t\}$, $Y = \{y_1, \ldots, y_t\}$, and $E \subseteq X \times Y$, a *perfect matching* is a set $J \subseteq E$ such that $|J| = t$, and all elements in X and Y are covered by J. In a #PERFECT-MATCHING instance, we are asked to compute the number of all perfect matchings. Given any #PERFECT-MATCHING instance X, Y, and E, we construct the following instance of computing the winning probability of a given candidate for LotThenBorda.

Candidates: $\mathcal{C} = \{c, b\} \cup X \cup Y \cup A$, where $A = \{a_1, \ldots, a_{2t}\}$. Let $k = 2t$. Suppose ties are broken in the following order: $X \succ Y \succ c \succ$ Others. We are asked to compute the probability that c wins.

Profile: For each edge $(x_i, y_j) \in E$, we first define a vote $W_{i,j} = [X \succ a_i \succ c \succ Y \succ b \succ \text{Others}]$, where elements within Y, X, A_i and B_j are ranked in ascending order of their subscripts. Then, we obtain $V_{i,j}$ from $W_{i,j}$ by exchanging the positions of the following two pairs of candidates: (1) x_i and a_i; (2) y_j and b. Let $P_V = \{V_{i,j} : \forall (x_i, y_j) \in E\}$.

For each $j \leq t$, we define a vote $U_j = [\text{rev}(Y) \succ c \succ a_{t+j} \succ \text{rev}(X) \succ \text{Others}]$, where rev$(X)$ is the linear order where the candidates in X are ranked in descending order of their subscripts. Let $P_U = \{U_1, \ldots, U_t\}$. Let the profile be $P = P_V \cup P_U$.

Let P' be a sub-profile of P such that $|P'| = k = 2t$. We first claim that if Borda$(P') = c$, then $P_U \subseteq P'$. For the sake of contradiction, suppose $P_V \cap P' = \{V_{i_1,j_1}, \ldots, V_{i_l,j_l}\}$, where $l > t$. Because $|X| = t$, there exists $i \leq t$ such that i is included in the multiset $\{i_1, \ldots, i_l\}$ at least two times. For any candidate c', let $s(P, c')$ denote the Borda score of c' in P. It follows that $s(P, x_i) > s(P, c)$, which contradicts the assumption that c is the Borda winner.

Next, we prove that for any $P' = P_U \cup \{V_{i_1,j_1}, \ldots, V_{i_t,j_t}\}$ such that Borda$(P') = c$, $J = \{(x_{i_1}, y_{j_1}), \ldots, (x_{i_t}, y_{j_t})\}$ is a perfect matching. Suppose J is not a perfect matching. If $x \in X$ (respectively, $y \in Y$) is not covered by J, then we have $s(P, x) = s(P, c)$ (respectively, $s(P, y) = s(P, c)$), which means that c is not the Borda winner due to tie-breaking. This contradicts the assumption. We note that different P' correspond to different perfect matchings. Similarly, any per-

fect matching corresponds to a different profile P' such that $|P'| = 2t$ and Borda$(P') = c$. We note that the probability that c wins is the number of such P' divided by $\binom{t+|E|}{2t}$. Therefore, computing the probability for c to win is #P-hard. It is easy to check that computing the probability for c to win is in #P. □

6 Manipulation

Suppose there are a group of k manipulators, who know the vote of the non-manipulators. There are at least three different dimensions to an analysis of manipulation in lot-based voting rules. The first two dimensions are standard, and the third dimension is specific for the lot-based rules.

The first dimension: weighted or unweighted votes.

The second dimension: constructive or destructive. Given a positive number p, in constructive manipulations, the manipulators seek to cast votes to make a given candidate win with probability at least p; in destructive manipulations, the manipulators seek to cast votes to make a given candidate lose with probability at least p.

The third dimension: fixed or adaptive. The manipulation is fixed, if all agents must declare a fixed preference ordering in advance of the lottery. In particular, the manipulators are not allowed to change their votes after lots are drawn. The manipulation is adaptive, if the manipulators observe the drawing of lotteries and can change their votes in light of which agents remain in the electoral college after the lottery. An adaptive manipulation is then described in terms of a strategy.

In this paper, we consider the manipulation problem where we are also given a positive number $p \leq 1$ and we are asked whether the manipulators can make a favored candidate c win with probability strictly larger than p. We stress that we are not asked how to compute the optimal strategy for the manipulators. These manipulation problems are closely related. For example, if fixed manipulation is possible for some p then adaptive manipulation is also possible for at least the same p. The same strategic vote will ensure this. However, the problems have different computational complexities. Whilst fixed manipulation is in NP, it is not immediately obvious that adaptive manipulation is even in PSPACE. In general, adaptive manipulations seem to be harder to compute than fixed manipulations. However, surprisingly, there are (somewhat artificial) lot-based voting rules where adaptive manipulation is easy to compute but fixed manipulation is intractable.

Theorem 8. *When the number of candidates is unbounded, there exists an instance of LotThen$X$ for which unweighted adaptive constructive manipulation is polynomial for any size of coalition, but unweighted fixed constructive manipulation is* NP*-hard for even a single manipulator.*

Proof sketch: We will use the 1-in-3-HittingSet (denoted by 1-IN-3HS) problem in this proof, which is known to be NP-complete [17]. In a 1-IN-3HS instance, we are given a set of Boolean variables $\mathcal{V} = \{\mathbf{x}_1, \ldots, \mathbf{x}_q\}$, and a set of t positive clauses $\mathcal{S} = \{S_1, \ldots, S_t\}$, where for each $j \leq t$, $S_j \subseteq \mathcal{V}$ and $|S_j| \leq 3$, that is, S_j contains at most 3 positive literals. We are asked whether there exists a valuation for $\mathcal{V}$ such that for every $j \leq t$, exactly one of the positive literals in S_j is satisfied.

We consider a lottery that picks two votes at random and the following rule X on two votes: the rule always selects either c_1 or c_2. If one vote has c_1 on top, the other vote has c_2 on top, and the vote with c_1 on top encodes a 1-IN-3HS satisfying assignment to the positive clause encoded by the vote with c_2 on top, then the winner is c_2. In any other situations, c_1 wins. To encode a truth assignment within a vote, we let $m = 2l+2$, and for each $i \leq l$, c_{2i+1} is ranked above c_{2i+2} if and only if X_i is true; to encode a positive clause within a vote, for each $i \leq l$, c_{2i+1} is ranked above c_{2i+2} if and only if X_i is in the clause.

Adaptive manipulation is now polynomial to compute since, for any lot containing a manipulator, we can easily compute whether the manipulators can make c_2 (c_1) win, and for any lot not containing a manipulator, we can also easily compute the winner. Thus, we can easily compute the maximum probability with which c_2 (c_1) can be made to win (and a manipulation that will achieve any probability up to this maximum). On the other hand, consider a fixed manipulation problem with a single manipulator in which the votes of the non-manipulators rank c_2 on top, and their votes encode the $t = n - 1$ positive clauses in a 1-IN-3HS instance. The only chance that c_2 can be made to win is when the manipulator is drawn in the random lot and votes with a "satisfying assignment". The probability that the random lot contains the manipulator is $\frac{2}{n}$. Hence, computing a fixed constructive manipulation for c_2 and $p = \frac{2}{n} - \frac{1}{\binom{n}{2}}$ is equivalent to finding a 1-IN-3HS satisfying assignment to all t clauses. □

7 Sampling the runoff voters non-uniformly

So far we have not discussed in details how to select the runoff voters. Of course if we only need to select k voters uniformly at random, then we can perform a naïve k-round sampling: in each round, a voter is drawn uniformly at random from the remaining voters, and is then removed from the list. However, it is not clear how to generate k voters with some non-uniform distribution. For example, different voters in a profile may have different voting power [16], and we may therefore want to generate the voters in the runoff according to this voting power. More precisely, we want to compute a probability distribution over all sets of k voters, and each time we randomly draw a set (of k voters) according to this distribution to meet some constraints. Let $\mathcal{M}$ denote the set of all $n \times k$ 0-1 matrices, in each of which the sum of each row is no more than 1 and the sum of each column is exactly 1. That is, $\mathcal{M} = \{(a_{(i,j)}) : a_{(i,j)} \in \{0,1\}, \forall i \leq n, \sum_j a_{(i,j)} \leq 1$ and $\forall j \leq k, \sum_i a_{(i,j)} = 1\}$. Each matrix in $\mathcal{M}$ represents a set of k voters. Formally, we define the sampling problem as follows.

Definition 3. *In a* LOTSAMPLING *problem , we are given a natural number n (the number of initial voters), a natural number k (the number of runoff voters), and a vector of positive real numbers $(p_1, \ldots, p_n)$ such that for any $j \leq n$, $0 \leq p_j \leq 1$ and $\sum_{j \leq n} p_j = k$. We are asked to compute a sampling technique that chooses k voters each time, and for every $j \leq n$, the probability that vote j is chosen is p_j.*

To solve the LOTSAMPLING problem, we first solve the following equations.

$$\forall i \leq n, \sum_j x_{(i,j)} = p_i \quad \text{and} \quad \forall j \leq k, \sum_i x_{(i,j)} = 1 \qquad (1)$$

We note that $\sum_{i \leq n} p_i = k$. For such equations, a solution where $x_{(i,j)} \geq 0$ for all $i \leq n, j \leq k$ always exists. To see this, we construct the solution by a greedy algorithm. The algorithm tries to settle the values row after row, and in each row, it tries to place as much "mass" as possible to the leftmost variable, as long as it does not violate the column constraints. Algorithm 1 does this.

Proposition 1. *Algorithm 1 returns a solution to Equations (1). Moreover, the number of non-zero entries in $(x_{(i,j)})$ is no more than $n + k$.*

Let $(x_{(i,j)})$ denote the outcome of Algorithm 1. Since the number of non-zero entries in $(x_{(i,j)})$ is no more than $n + k$, we can apply any polynomial-time algorithm that implements the Birkhoff-von Neumann theorem [4][5] to obtain a probability distribution over the matrices in $\mathcal{M}$ such that (1) the expectation is $(x_{(i,j)})$, and (2) the support of the distribution has no more than $n + k$ elements. That is, even though $|\mathcal{M}|$ is exponential, we only need to sample over a polynomial number of elements in $\mathcal{M}$. Therefore, we have the following theorem.

Theorem 9. *The* LOTSAMPLING *problem always has a solution that runs in polynomial-time.*

Algorithm 1: SolveEquation

Input: $(p_1, \ldots, p_n)$, where $\sum_j p_j = k$ and $\forall j \leq n$, $0 \leq p_j \leq 1$.
Output: A solution to Equations (1).

Let $x_{(i,j)} = 0$, $J = 1$.;
for $l = 1$ *to* n **do**
  if $\sum_{i<l} x_{(i,J)} + p_l \leq 1$ **then**
    Let $x_{(l,J)} = p_l$.
  end
  else
    Let $x_{(l,J)} = 1 - \sum_{i<l} x_{(i,J)}$,
    $x_{(l,J+1)} = p_l - x_{(l,J)}$, and $J = J + 1$.
  end
end
return $(x_{i,j})$.

8 Future work

Lot-based voting seems worth further attention. There are many directions for future work in addition to the many questions already raised in this note. For instance, we could consider the computational complexity of EVALUATION for other lot-based voting rules. We could also consider the control of lot-based voting by the chair. In addition to the usual forms of control like addition of candidates or of voters, we have another interesting type of control where the chair chooses the outcome of the lottery. Such control is closely related to control by deletion of voters. Other types of control include choosing the size of the lottery and choosing the voting rule used after the lottery. Another interesting direction would be to consider the computation of possible and necessary winners under lot-based voting. Finally, it would be interesting to consider formal properties of the Doge rule.

[5] For example, the Dulmage-Halperin algorithm [9].

Acknowledgements

Toby Walsh is supported by the Australian Department of Broadband, Communications and the Digital Economy, the ARC, and the Asian Office of Aerospace Research and Development (AOARD-104123). Lirong Xia acknowledges a James B. Duke Fellowship and Vincent Conitzer's NSF CAREER 0953756 and IIS-0812113, and an Alfred P. Sloan fellowship for support.

References

[1] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. In *Proc. STOC*, pages 684–693, 2005.

[2] Noga Alon. Ranking tournaments. *SIAM Journal of Discrete Mathematics*, 20:137–142, 2006.

[3] John Bartholdi, III, Craig Tovey, and Michael Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[4] Garrett Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumn Rev, Ser. A, no. 5*, pages 147–151, 1946.

[5] Vincent Conitzer and Tuomas Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proc. IJCAI*, pages 781–788, 2003.

[6] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proc. EC*, pages 82–90, 2006.

[7] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *JACM*, 54(3):1–33, 2007.

[8] Oliver Dowlen. Sorting out sortition: A perspective on the random selection of political officers. *Political Studies*, 57:298–315, 2009.

[9] L. Dulmage and I. Halperin. On a theorem of Frobenius-Konig and J. von Neumann's game of hide and seek. *Trans. Roy. Soc. Canada III*, 49:23–29, 1955.

[10] Edith Elkind and Helger Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Proc. ISAAC*, 2005.

[11] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53:74–82, 2010.

[12] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[13] Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45:665–681, 1977.

[14] Marji Lines. Approval voting and strategy analysis: A Venetian example. *Theory and Decision*, 20:155–172, 1986.

[15] Miranda Mowbray and Dieter Gollmann. Electing the doge of venice: Analysis of a 13th century protocol. In *Proc. IEEE CSF*, pages 295–310, 2007.

[16] David M. Pennock and Lirong Xia. Voting power, hierarchical pivotal sets, and random dictatorships. To be presented at *WSCAI*, 2011.

[17] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proc. STOC*, pages 216–226, 1978.

[18] Arthur M. Wolfson. The ballot and other forms of voting in the italian communes. *The American Historical Review*, 5(1):1–21, 1899.

Possible Winners in Noisy Elections

Krzysztof Wojtas
AGH University of Science and Technology, Kraków, Poland

Piotr Faliszewski
AGH University of Science and Technology, Kraków, Poland

Abstract

Predicting election winners (or, election possible winners) is an important topic in computational social choice. Very generally put, we consider the following setting: There is some set of candidates C and some set of voters V (with preferences over C). We either do not know which candidates will take part in the election or we do not know which voters will cast their votes. However, for each set $C' \subseteq C$ (each set $V' \subseteq V$) we know probability $P_C(C')$ that exactly candidates in C' participate in the election (probability $P_V(V')$ that exactly voters in V' cast their votes). Our goal is to compute the probability that a given candidate $c \in C$ wins the election. In its full generality—with unrestricted probability distributions P_C and P_V—these problems can very easily become computationally hard. We provide natural restrictions on P_C and P_V that allow us to obtain positive results for several election systems, including plurality, approval, and Condorcet's rule. On the technical side, our problems reduce to counting solutions to the problems of election control.

1 Introduction

Predicting election winners is always an exciting activity: Who will be the new president? Will the company merge with another one? Will taxes be higher or lower? Naturally, predicting winners is a hard task, full of uncertainties. For example, we typically are not sure which voters will eventually cast their votes and, sometimes, even the set of available candidates may be uncertain (consider, e.g., a candidate withdrawing due to personal reasons). Further, typically we do not have complete knowledge regarding each voters' preference order.

Nonetheless, to optimize their behavior, agents involved in an election try to somehow tackle the winner prediction problem. To model imperfect knowledge regarding voters' preferences, Konczak and Lang [2005] introduced the possible winner problem (further studied by many other researchers; see, e.g., [Xia and Conitzer, 2008; Betzler and Dorn, 2009; Bachrach *et al.*, 2010]. In this paper we focus on a different type of uncertainty: We consider settings where the set of participating candidates and the set of voters are uncertain. (However, we do assume perfect knowledge regaring voters' preferences.)

Specifically, we study the following setting. We are given a voting rule, a set C of m candidates, and a set V of n voters (for each voter we have perfect knowledge as to how she would vote). We consider two possible scenarios:

1. The set of candidates is fixed, but for each set of voters V', $V' \subseteq V$, we have probability $P_V(V')$ that exactly the voters from V' show up for the vote.
2. The set of voters is fixed, but for each set of candidates C', $C' \subseteq C$, we have probability $P_C(C')$ that exactly the candidates from C' participate in the election.

Our goal is to compute, for each candidate $c \in C$, the probability that c is a winner.

Naturally, our task would very quickly become computationally prohibitive (or, difficult to represent on a computer) if we did not assume anything about P_C and P_V. We use the following restrictions: First, we assume that both P_C and P_V are polynomial-time computable. Second, we would like to assume that for each subset V' of voters (each subset C' of candidates) the value $P_V(V')$ (the value $P_C(C')$) depends only on the cardinality of V' (only on the cardinality of C'). In other words, we have a probability distribution regarding the number of voters (the number of candidates) participating in the election, but each same-cardinality subset is equally likely.

However, this second assumption is slightly too strong. Often, we may have additional knowledge regarding the nature of possible changes in the candidate/voter set. For example, the rules may be such that after a given point of time candidates can withdraw from the election but no new candidates can register. Similarly, we may know that some votes have already been cast and cannot be withdrawn. Thus, we refine our model to be the following: We start with some candidates and voters already in the election and we ask for the probability that a given candidate wins assuming that some random number of voters/candidates is added/deleted.

Formally, it turns out that our winner prediction setting reduces to the counting variants of election control problems; computational study of election control problems was initiated by Bartholdi, Tovey, and Trick [1992] and was continued by Hemaspaandra, Hemaspaandra, and Rothe [2007],

Meir et al. [2008], Erdélyi, Nowak, and Rothe [2009], Faliszewski, Hemaspaandra, and Hemaspaandra [2011], and others (see the survey of Faliszewski, Hemaspaadra, and Hemaspaandra, al. [2010]). However, to the best of our knowledge, this is the first paper to study counting variants of election control. (However, we should mention that Bachrach et al. [2010] consider counting variants of possible-winner problems. Nonetheless, their model and motivation are different from ours; they assume the set of voters is fixed, but the voters are unsure as to how to vote. We assume the voters are certain about their votes, but unsure about participation in the election. The resulting technical problem is very different. Somewhere in the middle between these two approaches is the model of [Hazon *et al.*, 2008], where each voter has a probability distribution among several possible votes.)

Our results are very preliminary. Following Hemaspaandra, Hemaspaandra, and Rothe [2007], we focus on three, quite different in spirit, voting rules: plurality, Condorcet's rule, and approval voting. Our results show that counting variants of constructive control by adding/deleting candidates/voters for these voting rules are polynomial-time solvable whenever the decision variants are. This means that for the respective cases our winner prediction problems are polynomial-time solvable.

The paper is organized as follows. In Section 2 we formally define elections, the voting rules that we study, and provide brief background on complexity theory (focusing on counting problems). In Section 3 we formally define counting variants of election control problems and link them to the winner prediction scenarios that motivate our work. Section 4 contains our technical results. We conclude in Section 5.

2 Preliminaries

Elections and Voting Systems. An *election* E is a pair (C, V) such that C is a finite set of candidates and V is a finite collection of voters. We typically use m to denote the number of candidates and n to denote the number of voters. Each voter has a preference order in which he or she ranks candidates from the most desirable one to the most despised one. For example, if $C = \{a, b, c\}$ and a voter likes b most and a least, then this voter would have preference order $b > c > a$. (However, under approval voting, instead of ranking the candidates each voter simply indicates which candidates he or she approves of.)

A *voting system* is a rule which specifies how election winners are determined. We allow an election to have more than one winner, or even to not have winners at all. This is natural as votes may provide inadequate information for a voting system to always pick a single winner (e.q., due to symmetry or due to the fact that a voting rule is so restrictive as to require some sort of a consensus for a decision to be made). In real-life elections there are elaborate rules for dealing with such situations. Here we disregard tie-breaking rules by focusing on the so-called unique winner model (see the next section). However, we point the reader to [Obraztsova *et al.*, 2011] for a discussion regarding the influence of tie-breaking for the case of election manipulation problem.

Let $E = (C, V)$ be an election. For each candidate $c \in C$, we define c's plurality score $score_E^p(c)$ to be the number of voters in V that rank c first. Candidates with highest plurality scores are plurality winners. Under approval voting, the score of candidate $c \in C$, $score_E^a(c)$, is the number of voters that approve of c. Again, candidates with highest scores are winners.

Another, perhaps more involved, election system is *Condorcet's rule*, in which a candidate $c \in C$ is a winner if and only if for each $c' \in C \setminus \{c\}$, more than half of the voters prefer c to c'. There can be at most one winner under Condorcet's rule and he or she is called the *Condorcet winner*. We write $N_E(c, c')$ to denote the number of voters in V that prefer c to c'; c is a Condorcet winner exactly if $N_E(c, c') > N_E(c', c)$ for each $c' \in C \setminus \{c\}$.

Computational Complexity. We assume that the reader is familiar with the basic notions of complexity theory, including such notions as NP and NP-completeness. Let us, however, briefly review notions regarding the complexity theory of counting problems. Let A be some computational problem where, for each instance I, we ask if there exists some mathematical object satisfying a given condition. In the counting variant of A, denoted $\#A$, we ask how many such mathematical objects exist. For example, consider the following definition.

Definition 1. *An instance of X3C is a pair $(B, \mathcal{S})$, where $B = \{b_1, \ldots, b_{3k}\}$ and $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a family of 3-element subsets of B. In X3C we ask if it is possible to find exactly k sets in $\mathcal{S}$ whose union is exactly B. In #X3C we ask how many k-element subsets of $\mathcal{S}$ have B as their union.*

The class of counting variants of NP-problems is called #P. To reduce counting problems to each other, we use the notion of a parsimonious reduction.

Definition 2. *Let $\#A$ and $\#B$ be two counting problems. We say that $\#A$ parsimoniously reduces to $\#B$ if there exists a polynomial-time computable function f such that for each instance I of $\#A$ the following two conditions hold:*

1. *$f(I)$ is an instance of $\#B$, and*
2. *I has exactly as many solutions as $f(I)$.*

We say that a problem is #P-parsimonious-complete if it belongs to #P and every #P-problem parsimoniously reduces to it. For example, #X3C is #P-parsimonious-complete. Throughout this paper we will write #P-complete to mean #P-parsimonious-complete. We should mention, however, that different authors sometimes use different reduction types to define #P-completeness. For example, Valiant [1979] used Turing reductions, Zankó [1991] used many-one reductions, and Krentel [1988] used metric reductions.

The class of functions computable in polynomial time is called FP. Thus, if a given counting problem can be solved in polynomial time then we will write that it is in FP.

3 Counting Variants of Control Problems

Let us now formally define the counting variants of the election control problems. We are interested in four types of control: control by adding candidates (AC), control by deleting candidates (DC), control by adding voters (AV), and control

by deleting voters (DV). For each of the problems we consider its constructive variant (CC) and its destructive variant (DC). We now formally define the counting variant of constructive control by adding voters and then explain informally how the counting variants of other control problems are defined. As is typical for computational study of control problems, we assume the *unique-winner* model.

Definition 3. *Let R be a voting system. In the counting variant of constructive control by adding voters problem for R (R-#CCAV) we are given a set of candidates C, a set of registered voters V, a set of unregistered voters W, a designated candidate $p \in C$, and a natural number k. We ask how many sets W', $W' \subseteq W$, are there such that p is the unique winner of R-election $(C, V \cup W')$, where $|W'| \leq k$.*

Constructive control by deleting voters (#CCDV) is defined analogously, but we do not have W in the input and we ask how many sets V', $V' \subseteq V$, are there such that p is the unique R-winner of $(C, V \setminus V')$ and $V' \leq k$.

In the constructive control by adding candidates (#CCAC) and the constructive control by deleting candidates (#CCDC) problems the set of voters is fixed but we can vary the set of candidates. In #CCAC we are given an additional set A of unregistered candidates and we ask for how many sets $A' \subseteq A$ of size up to k it holds that p is the unique winner of election $(C \cup A', V)$ (naturally, we assume that the voters have preferences over all candidates in $C \cup A$). In #CCDC we ask how many subsets C' of C are there of size up to k such that p is the unique winner of election $(C \setminus C', V)$.[1]

Destructive variants of our problems are defined analogously, except that we ask for the number of settings where the designated candidate—who in this case is called the despised candidate—is not the unique winner of the election.

Counting variants of control problems are interesting in their own right, but we focus on them because they allow us to model winner prediction problems for settings where the structure of the election is uncertain. We now describe one example scenario, pertaining to #CCAV; the reader can imagine analogous settings for the remaining types of control.

Let us assume that set C of candidates participating in the election is fixed (for example, because the election rules force all candidates to register well in advance). We know that some set V of voters will certainly vote (for example, because they have already voted and this information is public[2]). The set of voters who have not decided to vote yet is W. From some source (e.g., from prior experience) we have some probability distribution P on the number of voters from W that will participate in the election (from our perspective, each equal-sized subset of voters from W is equally likely; different-sized sets may, of course, have different probabilities of participating in the election).

In other words, for each i, $0 \leq i \leq |W|$, let $P(i)$ be the probability that i voters from W join the election (and assume that we have an easy way of computing this value) and let $Q(i)$ be the probability that a designated candidate p wins under the condition that exactly i voters from W participate (assuming that each i-element subset of W is equally likely). Then, the probability that p wins is simply given by:

$$P(0)Q(0) + P(1)Q(1) + \cdots P(|W|)Q(|W|).$$

To compute $Q(i)$, we have to compute for how many sets W, of size exactly i, candidate p wins, and divide it by $\binom{|W|}{i}$. To compute for how many sets of size exactly i candidate p wins, we solve the corresponding #CCAV problem for adding at most i voters from W, then for adding at most $i-1$ voters from W, and then we subtract the results.

[1] Formally, we forbid C' from containing p. In the constructive setting this follows from the definition but in the destructive one we have to assume it separately.

[2] Naturally, in typical political elections such information would not be public and we would have to rely on polls. However, in multiagent systems there can be cases where votes are public.

Problem	Plurality	Approval	Condorcet
#CCAC	#P-com	–	–
#DCAC	#P-com	FP	FP
#CCDC	#P-com	FP	FP
#DCDC	#P-com	–	–
#CCAV	FP	#P-com	#P-com
#DCAV	FP	?	?
#CCDV	FP	#P-com	#P-com
#DCDV	FP	?	?

Table 1: The complexity of counting variants of control problems. A dash in an entry means that the given system is *immune* to the type of control in question (i.e., it is impossible to achieve the desired effect by the action this control problem allows; technically this means the answer to the counting question is always 0). Immunity results were established by Bartholdi, Tovey, and Trick [1989] for the constructive cases and by Hemaspaandra, Hemaspaandra, and Rothe [2007] for the destructive cases. For the cases of #DCAV and #DCDV under approval voting and under Condorcet voting, we were able to show #P-metric-completeness but not #P-parsimonious-completeness.

4 Results

In this section we present our complexity results regarding counting variants of election control problems, focusing on positive, algorithmic results. We present a summary of our results in Table 1. In all constructive cases where a decision variant of a given problem is polynomial-time solvable, so is the counting variant. In all cases where a decision variant of a given problem is NP-complete, the counting variant is #P-complete. We do not present our #P-completeness proofs/theorems as they are mostly easy extensions of the constructions already present in the literature. Our #P-completeness results follow by reductions from #X3C.

4.1 Plurality Voting

Under plurality voting, counting variants of both control by adding voters and control by deleting voters are in FP. In both cases our algorithms are based on dynamic programming. We believe that our approach can be used for several other voting systems.[3]

[3] Most glaring example of such a rule would be veto. For example, under veto adding voters is essentially the same as deleting

Theorem 4. *Plurality-#CCAV is in* FP.

Proof. Let $I = (C, V, W, p, k)$ be an input instance of Plurality-#CCAV, where $C = \{p, c_1, \ldots, c_{m-1}\}$ is the candidate set, V is the set of registered voters, W is the set of unregistered voters, p is the designated candidate, and k is the upper bound on the number of voters that can be added. We now describe a polynomial-time algorithm that computes the number of solutions for I.

Let A_p be the set of voters from W that rank p first. Similarly, for each $c_i \in C$, let A_{c_i} be the set of voters from W that rank c_i first. We also define $count(C, V, W, p, k, j)$ to be the number of sets $W' \subseteq W - A_p$ such that:

1. $|W'| \leq k - j$, and
2. in election $(C, V \cup W')$ each candidate $c_i \in C, 1 \leq i \leq m-1$, has score at most $score^p_{(C,V)}(p) + j - 1$.

The pseudocode for our algorithm is given below.

PLURALITY-#CCAV(C, V, W, p, k)
1 **if** p is the unique winner of (C, V)
2 **then** $k_0 := 0$
3 **else** $k_0 := \max_{c_i \in C}(score^p_{(C,V)}(c_i) - score^p_{(C,V)}(p) + 1)$,
4 $result := 0$
5 **for** $j := k_0$ **to** $\min(|A_p|, k)$
6 **do** $result := result + \binom{|A_p|}{j} \cdot count(C, V, W, p, k, j)$
7 **return** $result$

At the beginning, the algorithm computes k_0, the minimum number of voters from A_p that need to be added to V to ensure that p has plurality score higher than any other candidate (provided no other voters are added). Clearly, if p already is the unique winner of (C, V) then k_0 is 0, and otherwise k_0 is $\max_{c_i \in C}(score^p_{(C,V)}(c_i) - score^p_{(C,V)}(p) + 1)$. After we compute k_0, for each j, $k_0 \leq j \leq \min(k, |A_p|)$, we compute the number of sets W', $W' \subseteq W$, such that W' contains exactly j voters from A_p, at most $k - j$ voters from $W - A_p$, and p is the unique winner of $(C, V \cup W')$. It is easy to verify that for a given j, there is exactly $h(j) = \binom{|A_p|}{j} \cdot count(C, V, W, p, k, j)$ such sets. Our algorithm returns $\sum_{j=k_0}^{\min(k,|A_p|)} h(j)$. The reader can easily verify that this indeed is the correct answer. To complete the proof it suffices to show a polynomial-time algorithm for computing $count(C, V, W, p, k, j)$.

Let us fix j, $k_0 \leq j \leq \min(k, |A_p|)$ and show how to compute $count(C, V, W, p, k, j)$. Our goal is to count the number of ways in which we can add at most $k - j$ voters from $W - A_p$ so that no candidate $c_i \in C$ has score higher than $score^p_{(C,V)}(p) + j - 1$. For each candidate $c_i \in C$, we can add at most

$$l_i = \min\big(|A_{c_i}|, j + score^p_{(C,V)}(p) - score^p_{(C,V)}(c_i) - 1\big),$$

voters from A_{c_i}; otherwise c_i's score would exceed $score^p_{(C,V)}(p) + j - 1$.

voters under plurality.

For each i, $1 \leq i \leq m-1$, and each t, $0 \leq t \leq k-j$, let $a_{t,i}$ be the number of sets $W' \subseteq A_{c_1} \cup A_{c_2} \cup \cdots \cup A_{c_i}$ that contain exactly t voters and such that each candidate $c_1, c_2, \ldots, c_i$ has score at most $score^p_{(C,V)}(p) + j - 1$ in the election $(C, V \cup W')$. Naturally, $count(C, V, W, p, k, j) = \sum_{t=0}^{k-j} a_{t,m-1}$. It is easy to check that $a_{t,i}$ satisfies the following recursion:

$$a_{t,i} = \begin{cases} \sum_{s=0}^{\min(l_i,t)} \binom{|A_{c_i}|}{s} a_{t-s,i-1}, & \text{if } t > 0, i > 1, \\ 1, & \text{if } t = 0, i > 1, \\ \binom{|A_1|}{t}, & \text{if } t \leq |A_{c_1}|, i = 1, \\ 0, & \text{if } t > |A_{c_1}|, i = 1. \end{cases}$$

Thus, for each t, i we can compute $a_{t,i}$ using standard dynamic programming techniques in polynomial time. Thus, $count(C, V, W, p, k, j)$ also is polynomial-time computable. This completes the proof. □

Using this algorithm, we can easily derive one for the destructive setting.

Theorem 5. *Plurality-#DCAV is in* FP.

Proof. Let $I = (C, V, W, p, k)$ be an instance of plurality-#CCAV. There are exactly $\sum_{i=0}^{k} \binom{|W|}{i}$ sets W' such that $W' \subseteq W$ and $|W'| \leq k$. Of these, there are exactly PLURALITY-#CCAV(C, V, W, p, k) sets of voters whose inclusion in the election ensures that p is the unique winner. Thus, there are exactly

$$\sum_{i=0}^{k} \binom{|W|}{i} - \text{PLURALITY-\#CCAV}(C, V, W, p, k)$$

subsets of W of cardinality at most k whose inclusion in the election ensures that p is not the unique winner. Clearly, we can compute this value in polynomial time. □

Given the results for control by adding voters, it is not surprising that similar results hold for the case of deleting voters.

Theorem 6. *Plurality-#CCDV is in* FP.

Proof. Let $I = (C, V, p, k)$ be an instance of plurality-#CCDV, where $C = \{p, c_1, \ldots, c_{m-1}\}$ is the set of candidates, V is the set of voters, p is the designated candidate, and k is the upper bound on the number of voters that can be deleted. We will now give a polynomial-time algorithm that computes the number of solutions for I.

Let A_p be the subset of V containing those voters that rank p first. Similarly, for each $c_i \in C$, let A_{c_i} be the subset of voters that rank c_i first. For each integer j, $0 \leq j \leq k$, we define $count(C, V, p, k, j)$ to be the number of subsets V' of $V - A_p$ such that:

1. $|V'| \leq k - j$, and
2. in election $(C, V - V')$ each candidate $c_i \in C$ has score at most $score^p_{(C,V)}(p) - j - 1$.

The algorithm given below returns the number of solutions for I.

PLURALITY-#CCDV(C, V, p, k)

1 $result := 0$
2 **for** $j := 0$ **to** $\min(|A_p|, k)$
3 **do** $result := result + \binom{|A_p|}{j} \cdot count(C, V, p, k, j)$
4 **return** $result$

In each iteration of the main loop we consider deleting exactly j voters from A_p (there are $\binom{|A_p|}{j}$ ways to pick these j voters) . Assuming we remove from V exactly j members of A_p, we must also remove some number of voters from $V - A_0$, to make sure that p is the unique winner of the resulting election. The number of ways in which this can be achieved is $count(C, V, p, k, j)$. It is easy to verify that indeed our algorithm works correctly. It remains to show how to compute $count(C, V, p, k, j)$.

Let us fix some value j, $0 \leq j \leq \min(k, |A_p|)$. We will show how to compute $count(C, V, p, k, j)$. For each $c_i \in C$, we define:

$$l_i = \max\big(0, j + score^p_{(C,V)}(c_i) - score^p_{(C,V)}(p) + 1\big).$$

Intuitively, l_i is the minimal number of voters from A_{c_i} that need to be removed from the election for p to have score higher than c_i (assuming j voters from A_p have been already removed from the election).

For each i, $1 \leq i \leq m-1$, and each t, $0 \leq t \leq k-j$, let $a_{t,i}$ be the number of sets $V' \subseteq A_{c_1} \cup A_{c_2} \cup \cdots \cup A_{c_i}$ such that $|V'| = t$ and each candidate $c_1, \ldots, c_i$ has score at most $score^p_{(C,V)}(p) - j - 1$ in election $(C, V - V')$. It is easy to see that $count(C, V, p, k, j) = \sum_{t=0}^{k-j} a_{t,m-1}$. Further, the following recursive relation holds:

$$a_{t,i} = \begin{cases} \sum_{s=l_i}^{\min(|A_{c_i}|,t)} \binom{|A_{c_i}|}{s} a_{t-s,i-1}, & \text{if } t \geq l_i,\ i > 1, \\ 0, & \text{if } t < l_i, \\ \binom{|A_1|}{t}, & \text{if } t \geq l_1,\ i = 1. \end{cases}$$

Thus, for each t, i we can compute $a_{t,i}$ in polynomial time using dynamic programming. As a result, we can compute $count(C, V, p, k, j)$ and the proof is complete. □

4.2 Approval Voting and Condorcet Voting

Let us now consider approval voting and Condorcet voting. While these two systems are very different in many respects, their behavior with respect to election control is very similar. Specifically, for both systems #CCAV and #CCDV are #P-complete, for both systems it is impossible to make some candidate a winner by adding candidates, and for both systems it is impossible to prevent someone from winning by deleting candidates. Yet, for both systems #DCAC and #CCDC are in FP via almost identical algorithms.

Theorem 7. *Both approval-#DCAC and Condorcet-#DCAC are in* FP.

Proof. We first consider the case of approval voting. Let $I = (C, A, V, p, k)$ be an instance of approval-#DCAC, where $C = \{p, c_1, \ldots, c_{m-1}\}$ is the set of registered candidates, $A = \{a_1, \ldots, a_{m'}\}$ is the set of additional candidates, V is the set of voters (with approval vectors over $C \cup A$), p is the designated candidate, and k is the upper bound on the number of candidates that we can add. We will give a polynomial-time algorithm that counts the number of up-to-k-element subsets A' of A such that p is not the unique winner of election $(C \cup A', V)$.

Let A_0 be the set of candidates in A that are approved by at least as many voters as p is. To ensure that p is not the unique winner of the election (assuming p is the unique winner prior to adding any candidates), it suffices to include at least one candidate from A_0. Thus, we have the following algorithm.

APPROVAL-#DCAC(C, A, V, p, k)

1 **if** p is not the unique winner of (C, V)
2 **then return** $\sum_{i=0}^{k} \binom{|A|}{i}$
3 Let A_0 be the set of candidates $a_i \in A$,
s.t. $score^a_{(C \cup A, V)}(a_i) \geq score^a_{(C \cup A, V)}(p)$.
4 $result := 0$
5 **for** $j := 1$ **to** k
6 **do** $result := result + \sum_{i=1}^{\min(|A_0|, j)} \binom{|A_0|}{i} \binom{|A - A_0|}{j-i}$
7 **return** $result$

The loop from line 5, for every j, counts the number of ways in which we can choose exactly j candidates from A; it can be done by first picking i of the candidates in A_0 (who beat p), and then $j - i$ of the candidates in $A - A_0$. It is clear that the algorithm is correct and runs in polynomial time.

Let us now move on to the case of Condorcet voting. It is easy to see that the same algorithm works correctly, provided that we make two changes: (a) in the first two lines, instead of testing if p is an approval winner we need to test if p is a Condorcet winner, and (b) we redefine the set A_0 to be the set of candidates $a_i \in A$ such that $N_{C \cup A}(p, a_i) \leq N_{C \cup A}(a_i, p)$. To see that these two changes suffice, it is enough to note that to ensure that p is not a Condorcet winner of the election we have to have that either p already is not a Condorcet winner (and then we can freely add any number of candidates), or we have to add at least one candidate from A_0. □

Theorem 8. *Both approval-#CCDC and Condorcet-#CCDC are in* FP.

Proof. Let us handle the case of approval voting first. Let $I = (C, V, p, k)$ be an instance of approval-#CCDC. The only way to ensure that $p \in C$ is the unique winner is to remove all candidates $c \in C - \{p\}$ such that $score^a_{(C,V)}(c) \geq score^a_{(C,V)}(p)$. Such candidates can be found immediately. Let's assume that there are k_0 such candidates. After removing all of them, we can also remove $k - k_0$ or less of any remaining candidates other than p. Based on this observation we provide the following simple algorithm.

APPROVAL-#CCDC(C, V, p, k)

1 Let k_0 be the number of candidates $c \in C - \{p\}$,
s.t. $score^a_{(C,V)}(c) \geq score_{(C,V)}(p)$.
2 **return** $\sum_{i=0}^{k-k_0} \binom{|C|-k_0-1}{i}$

Clearly, the algorithm is correct and runs in polynomial-time.

For the case of Condorcet voting, it suffices to note that if p is to be a winner, we have to delete all candidates $c \in C - \{p\}$ such that $N_{C,V}(p,c) \leq N_{C,V}(c,p)$. Thus, provided that we let k_0 be the number of candidates $c \in C - \{p\}$ such that $N_{C,V}(p,c) \leq N_{C,V}(c,p)$, the same algorithm as for the case of approval voting works for Condorcet voting. □

5 Conclusions and Future Work

We have considered a natural model of predicting election winners in settings where there is uncertainty regarding the structure of the election (that is, regarding the exact set of candidates and the exact collection of voters participating in the election). We have shown that our model corresponds to the counting variants of election control problems (specifically, we have focused on election control by adding/deleting candidates and voters).

Following the paper of Hemaspaandra, Hemaspaandra, and Rothe [2007], we have considered three voting rules: plurality, approval, and Condorcet voting. It turned out that the complexity of counting the number of solutions for constructive control problems under these systems is analogous to the complexity of verifying if any solution exists. That is, whenever the decision variant of the constructive problem is in P, the counting variant is in FP; whenever the decision variant is NP-complete, the counting variant is #P-complete. While the latter is not too surprising, sometimes easy decision problems correspond to hard counting problems, and thus the former is less trivial. However, perhaps this behavior is due to the simplicity of the election systems we have considered. Thus, the most natural research direction currently is to study counting variants of control under further election systems.

Currently, we are working on results for simplified variant of Dodgson and for maximin. The former is interesting because it is used to efficiently approximate the Dodgson rule [Caragiannis *et al.*, 2010]. The latter is interesting because it is known that several constructive control problems are easy for it [Faliszewski *et al.*, 2011]. A more involved research direction is to consider more involved probability distributions of candidates/voters that join/leave the election.

Acknowledgements. We are very grateful to WSCAI referees for helpful, thorough reports.

References

[Bachrach *et al.*, 2010] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *Proceedings of AAAI 2010*, pages 697–702, July 2010.

[Bartholdi *et al.*, 1989] J. Bartholdi, III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

[Bartholdi *et al.*, 1992] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[Betzler and Dorn, 2009] N. Betzler and B. Dorn. Towards a dichotomy of finding possible winners in elections based on scoring rules. In *Proceedings of MFCS 2009*, pages 124–136, August 2009.

[Caragiannis *et al.*, 2010] I. Caragiannis, C. Kaklamanis, N. Karanikolas, and A. Procaccia. Socially desirable approximations for Dodgson's voting rule. In *Proceedings of EC 2010*, pages 253–262, June 2010.

[Erdélyi *et al.*, 2009] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, 55(4):425–443, 2009.

[Faliszewski *et al.*, 2010] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[Faliszewski *et al.*, 2011] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[Hazon *et al.*, 2008] N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. Evaluation of election outcomes under uncertainty. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 959–966, May 2008.

[Hemaspaandra *et al.*, 1997] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[Hemaspaandra *et al.*, 2007] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[Konczak and Lang, 2005] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedins of the Multidisciplinary IJCAI-05 Worshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[Krentel, 1988] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[Meir *et al.*, 2008] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.

[Obraztsova *et al.*, 2011] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *Proceedings of AAMAS 2011*, 2011. To appear.

[Valiant, 1979] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Xia and Conitzer, 2008] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of AAAI 2008*, pages 196–201, July 2008.

[Zankó, 1991] V. Zankó. #P-completeness via many-one reductions. *International Journal of Foundations of Computer Science*, 2(1):76–82, 1991.

Consensus Action Games

Julian Zappala, Natasha Alechina, Brian Logan
School of Computer Science, University of Nottingham
{jxz,nza,bsl}@cs.nott.ac.uk

Abstract

We present Consensus Action Games (CAGs), a novel approach to modelling consensus action in multi-agent systems inspired by quorum sensing and other forms of decision making found in biological systems. In a consensus action game, each agent's degree of commitment to the joint actions in which it may participate is expressed as a quorum function, and an agent is willing to participate in a joint action if and only if a quorum consensus can be achieved by all the agents participating in the action.We study the computational complexity of several decision problems associated with CAGs and give tractable algorithms for problems such as determining whether an action is a consensus action. We briefly compare CAGs to related work such as Qualitative Coalitional Games.

1 Introduction

There are many reasons why agents may wish to, or indeed have to, cooperate, for example, where resources are constrained or otherwise in contention, where agents have differing abilities, or where they possess differing information. Even self interested agents may be motivated towards cooperative behaviour where this is consistent with individual rationality, for example, where cooperation increases their individual utility. While there has been considerable research in AI into joint actions and the collective execution of a shared plan [Levesque *et al.*, 1990; Grosz and Kraus, 1993; Tambe, 1997], the main focus of this work has been to examine how teams of autonomous agents may collectively achieve some goal. Relatively little attention has been paid to the selection of the joint actions that agents may perform.

However, the problem of collective action selection has been extensively studied in the fields of behavioural ecology and theoretical biology. In this paper, we propose a game-theoretic model which is an abstraction of several mechanisms for collective action selection occurring in nature.

Many natural systems, including bacteria [Jacob *et al.*, 2004], ants [Pratt *et al.*, 2005], bees [Seeley and Visscher, 2004], and fish [Ward *et al.*, 2008] exhibit a behaviour known as quorum sensing. Through a process termed the *quorum response*, the probability that an individual will select a particular action is increasing in the proportion of individuals already having made that choice. This relationship is typically non-linear such that the probability that an action is selected by an agent increases sharply once the number of agents that have already selected that action passes some threshold [Sumpter and Pratt, 2008]. The macroscopic behaviour of such a self-organising system resembles one in which individuals converge upon consensus with respect to a joint action. The prevalence of quorum decision making in nature suggests that this is an efficient, effective and stable mechanism through which group activities can be coordinated. Theoretical models support this view, predicting that where the quorum threshold is adaptive, decisions can not only be optimal [List, 2004] but also provide a trade-off between speed and accuracy [Pratt and Sumpter, 2006]. Conradt and Roper [2005] term this type of group decision *combined decisions*.

Another mechanism for collective action selection is found in spatially cohesive groups of (often social) animals, where decisions must be made regarding, e.g., movement direction, travel destination and activity timing. For example, a group of primates may need to decide whether to forage or go to a water source. To minimise the risk of predation, it is critical that whatever action is chosen is a consensus action, i.e., is performed jointly by all the agents, but each agent will typically have differing preferences for each joint action and for which other agents participate in the action (e.g., mutual grooming with a high status individual). The mechanisms by which such consensus actions are selected are poorly understood. However there is evidence from field observations to suggest that the more individuals indicate they are in favour of a particular action (for example, by making tentative moves in a particular direction), the more likely are the other animals to 'agree' to the action (see, for example [Stueckle and Zinner, 2008]). Conradt and Roper [2005] term this type of group decision *consensus decisions*.

In this paper we present *Consensus Action Games* (CAGs), a novel approach to modelling collective action selection in multi-agent systems inspired by mechanisms for reaching combined and consensus decisions in natural systems. In a consensus action game, each agent's degree of commitment to the joint actions in which it may participate is expressed as a quorum function, and its decision whether to support a joint action is mediated by the quorum thresholds of the

other agents that may participate in the action. Consensus is reached, where possible, through a series of individual commitments. We study the computational complexity of several decision problems associated with CAGs and give tractable algorithms for problems such as determining whether an action is a consensus action. We briefly compare CAGs to related work such as Qualitative Coalitional Games.

Although the immediate motivation for consensus action games are the phenomena underlying combined and consensus decisions in biological systems, we believe the model has wider application, for example, modelling trend adoption in people, and consensus action in multiagent systems. As such, it extends current work in coalition formation in multiagent systems, in considering not only which coalition an agent should join, but which action the agent performs as part of that coalition.

The remainder of this paper is organised as follows. In section 2 we introduce Consensus Action Games (CAGs), and in section 3 we consider the complexity of decision problems associated with CAGs. We discuss related research in section 4, and in section 5, we conclude and suggest directions for future work.

2 Consensus Action Games

A consensus action game (CAG) is a tuple $\Gamma = \langle G, A, J, q \rangle$ where:

G is a finite set of agents, $\{1, \ldots, n\}, n \geq 2$

A is a finite, non empty set of possible actions $\{1, \ldots, m\}$

J is a set of joint actions; each joint action is a set of pairs (i, a), where $i \in G$ and $a \in A$, specifying the action performed by each agent participating in the joint action. We write $J_i = \{j \in J \mid (i, a) \in j\}$ to indicate the set of joint actions in which agent i may participate, and $J_{G'} = \{j \in J \mid \{i \mid (i, a) \in j\} = G'\}$ for the set of all joint actions that can be performed by the set of agents $G' \subseteq G$.[1]

q is a quorum function which takes an agent $i \in G$ and an action j in J_i and returns a number in the interval [0,1], formally $q : \{(i, j) \mid i \in G, j \in J_i\} \rightarrow [0, 1]$. We will sometimes write $q_i(j)$ for $q(i, j)$. For an agent $i \in G' \subseteq G$ and joint action $j \in J_{G'}$, the quorum function $q_i(j)$ gives the minimum proportion of agents in G' which must support j in order that i will support j. Where $q_i(j) = 0$ agent i shows unconditional support for j, where $0 < q_i(j) < 1$ the agent shows conditional support for j; where $q_i(j) = 1$ the agent does not support j.

We say there is a *quorum consensus* about a joint action j if and only if all agents participating in j support j. Let $G' \subseteq G$, $j \in J_{G'}$, and $Q \subseteq G'$. Consider a function $Support_j : Q \mapsto Q \cup \{i \in G' \mid q_i(j) \times |G'| \leq |Q|\}$. Then the joint action j is a *quorum consensus action* if and only if G' is the least fixed point of $Support_j$. We will refer to each invocation of $Support_j$ as a *round*.

[1] Note that the set of joint actions is not simply the Cartesian product of all possible individual actions.

2.1 Example

Consider a group of six agents which have actions sing (s), play (p) and have a party (h). There are three joint actions:

$j_1 = \{(6, s), (2, p)\}$ with $q(6, j_1) = 0$ and $q(2, j_1) = 1/4$

$j_2 = \{(6, s), (3, p)\}$ with $q(6, j_2) = 0$ and $q(3, j_2) = 3/4$

$j_3 = \{(1, h), (2, h), \ldots, (6, h)\}$ with $q(i, j_3) = (i - 1)/6$

Intuitively, agent 6 is keen to sing, and agent 2 will consent to participate in the joint action j_1 where 6 sings and 2 plays accompaniment, because 2 requires at least a quarter of the agents involved in the action to support it before it declares its support, and agent 6 (half of the agents) supports it. Hence j_1 is a quorum consensus action. Action j_2 is not a quorum consensus action (agent 6 has unconditional support for it, but taking this into account only half of the agents support the action, and agent 3 requires three quarters). Finally, action 3 is a quorum consensus action: agent 1 has unconditional support for it, agent 2 supports it provided 1/6 of the agents do (which agent 1 does), agent 3 supports it if 2 out of 6 agents do (which 1 and 2 do), and so on. Observe that if we had $q(6, j_3) = 1$ rather than 5/6, then j_3 would not be a quorum consensus action.

The first two actions illustrate joint actions which are 'joint activities' (actions which require several participants to be performed) while the third action can be seen as somewhat similar to the quorum sensing in bacteria (all agents do the same thing, and the larger the number of agents that support the action, the larger the number of agents who are willing to participate in the action).

3 Computational Complexity of CAGs

Our characterisation of the computational complexity of consensus action games focuses on three natural decision problems associated with the selection of joint actions.

Consensus Action (CA): is an action a consensus action?

Group Consensus (GC): does a particular group of agents have a consensus action?

No Consensus (NC): is it the case that no group of agents has a consensus action?

We begin by considering the size of the input to the decision problems, namely the size of the representation of a CAG. Given a set of agents of size n and a set of actions of size m, in the worst case (when every set of agents can jointly execute any possible combination of actions) the set of joint actions J has cardinality $O(m^n)$, i.e., exponential in the number of agents. However, for any particular CAG $|J|$ may be significantly smaller than m^n.

We assume a concise representation of the input in which each joint action j is encoded as a set of triples rather than pairs: each triple consists of an agent, an action and the value of the quorum function for the agent and joint action. Thus q is encoded in J. We also assume the function $agents : J \rightarrow \mathfrak{P}(G)$ returns $G' \subseteq G$, the set of agents that may participate in action j, which runs in at most $O(n)$. Finally, we assume that J is implemented as a random access data structure and that we can determine the size (number of elements) in J in $O(log|J|)$.

The first three decision problems consider the complexity of determining CA, GC and NC for quorum consensus actions.

QUORUM CONSENSUS ACTION(QCA)

Given a CAG $\Gamma = \langle G, A, J, q\rangle$ and a joint action $j \in J$, is j a quorum consensus action?

Algorithm: The algorithm must verify that $agents(j)$ is the least fixed point of $Support_j$.

Time Complexity: $O(n)$.

Algorithm 1 Is j a quorum consensus action.

```
function QCA(j, Γ)
    array support[|j| + 1] ← {0, ..., 0}
    for all (i, a, q) ∈ j do
        k ← ⌈q × |j|⌉
        support[k] ← support[k] + 1
    s ← support[0]
    for k from 1 to |j| do
        if k ≤ s then
            s ← s + support[k]
        else
            return false
    return true
```

Note that we can also obtain an $O(n \times log(n))$ algorithm, which runs in constant space by sorting j.

QUORUM GROUP CONSENSUS (QGC)

Given a CAG $\Gamma = \langle G, A, J, q\rangle$ and a subset of agents $G' \subseteq G$, is there a quorum consensus action for G'?

Algorithm: The algorithm must verify that $\exists j \in J_{G'}$ such that G' is the least fixed point of $Support_j$.

Time Complexity: $O(n \times |J|)$

A non-deterministic algorithm first guesses an index of an action j in J (this can be done in $O(log(|J|)) \leq O(n)$ by the assumption that we can get the size of J in $O(log(|J|))$, and then checks that $agents(j) = G'$ and that j is a consensus action. This can be done in time linear in n using Algorithm 1. This gives us a non-deterministic linear time algorithm for a random access machine.[2] Hence, the problem is in NP(n) for RAM.

QUORUM NO CONSENSUS (QNC)

Given a CAG $\Gamma = \langle G, A, J, q\rangle$, is it the case that no subset $G' \subseteq G$ has a quorum consensus action?

Algorithm: The algorithm must verify that $\neg\exists j \in J$ such that G' is the least fixed point of $Support_j$.

Time Complexity: $O(n \times |J|)$

Since the problem of the existence of a quorum consensus action is in NP(n) for RAM (guess an action in J and verify it is a quorum consensus action), its complement QNC is in co-NP(n) for RAM.

[2] As Immerman [1998] has observed, such machines correspond more closely to real computers than do multi-tape Turing machines.

3.1 The Core of Consensus Action Games

The core is a key solution concept in game theory that aggregates stable outcomes which are both individually and collectively rational. In CAGs, agents are willing to participate in any joint action where the degree of support for the action exceeds the agent's quorum threshold. However, a rational agent will disregard joint actions in which not all agents are willing to participate as these are unlikely to be performed. Thus the only joint actions in which an agent would actually participate are consensus actions. Consensus actions are therefore individually rational and, in one sense, stable. Collectively rational outcomes are, traditionally, those where no subset of agents can find improvement through unilateral defection. We consider the complexity of two complimentary solution concepts for the core of CAGs.

G'-Minimal Consensus

Our first solution concept takes a similar approach to the qualitative model of the core introduced in [Wooldridge and Dunne, 2004]. We define the G'-minimal core of CAGs as containing only G'-minimal consensus actions. A G'-*minimal consensus action* is a quorum consensus action for which no subset $G'' \subset G'$ of agents have a quorum consensus action. The G'-minimal core aggregates quorum consensus actions which are collectively rational in the sense that they are immune to unilateral defection by some agents $G'' \subset G'$.

Below we consider the complexity of determining CA, GC and NC under the solution concept of the G'-minimal core.

G'-MINIMAL CONSENSUS ACTION (GMCA)

Given a CAG $\Gamma = \langle G, A, J, q\rangle$ and a joint action $j \in J$ by the agents $G' \subseteq G$, is j a G'-minimal consensus action for G'?

Algorithm: The algorithm must verify that G' is the least fixed point of $Support_j$ and that $\forall G'' \subset G', \neg\exists k \in J_{G''}$ such that G'' is the least fixed point of $Support_k$.

Time Complexity: $O(n \times |J|)$.

A non-deterministic algorithm to solve the complement of this problem (decide whether an action is *not* a G'-minimal consensus action) first checks whether j is a quorum consensus action (and returns true if it is not); if j is a quorum consensus action, it will guess an index of an action $k \in J$ and check that $agents(k) \subset G'$ and k is a quorum consensus action. So the problem of deciding whether an action is *not* a minimal quorum consensus action is in NP(n) on RAM. Hence deciding whether an action is a G'-minimal consensus action is in co-NP(n) for RAM.

G'-MINIMAL GROUP CONSENSUS (GMGC)

Given a CAG $\Gamma = \langle G, A, J, q\rangle$ and a subset of agents $G' \subseteq G$, is there a minimal quorum consensus action for G'?

Algorithm: The algorithm must verify that $\exists j \in J_{G'}$ such that G' is the least fixed point of $Support_j$ and that $\forall G'' \subset G', \neg\exists k \in J_{G''}$ such that G'' is the least fixed point of $Support_k$.

Time Complexity: $O(n \times |J|)$

A nondeterministic algorithm first calls an NP(n) oracle to check that G' has a quorum consensus action; if G' does have a quorum consensus action, it then calls an NP(n) oracle to

check whether any $G'' \subset G'$ has a quorum consensus action. Hence the problem is in $D^p(n)$ (on RAM).[3]

G'-MINIMAL NO CONSENSUS (GMNC)

Given a CAG $\Gamma = \langle G, A, J, q \rangle$, is it the case that no subset $G' \subseteq G$ has a G'-minimal consensus action?

Algorithm: The algorithm must verify that $\neg\exists j \in J$ by agents $G' \subseteq G$ s.t. G' is the least fixed point of $Support_j$ and that $\forall G'' \subset G', \neg\exists k \in J_{G''}$ such that G'' is the least fixed point of $Support_k$.

Time Complexity: $O(n \times |J|)$

Note that if any subgroup of agents has a quorum consensus action then either that joint action, or some joint action by a subset of those agents will be minimal.

A non-deterministic polynomial time algorithm on RAM for solving the *complement* of this problem (to accept CAGs with non-empty G'-minimal core) would guess an action in J and verify that it is a quorum consensus action. Hence the problem of deciding whether the G'-minimal core is empty is co-NP(n).

$q-$Minimal Consensus

Our second solution concept focuses on the difficulty of reaching consensus. We define the *q-minimal core* of a CAG as containing only those joint actions for which the number of rounds required for quorum consensus is minimal. Specifically, a quorum consensus action j by the agents G' is a *q-minimal consensus action* if there is no other quorum consensus action for G' where the number of rounds required to reach consensus is less than the number of rounds required to reach consensus regarding j.

Below we consider the complexity of determining CA, GC and NC under the solution concept of the q-minimal core. We begin by defining a function $rounds$ that computes the number of rounds before the least fixed point of $Support_j$ is encountered.

Algorithm 2 Number of consensus rounds for j.

```
function rounds(j)
    Q ← 0
    r ← 0
    i1 ← 0
    i2 ← −1
    sort(j) by ascending qi(j)
    for all (i, a, q) ∈ j do
        if q × |j| ≤ Q then
            Q ← Q + 1
            i1 ← ⌊(q × |j|)⌋
            if (i1 > i2) then
                r ← r + 1
                i2 ← i1
    return r
```

Algorithm 2 has time complexity of $O(n \times log(n))$.

[3]The Difference class is the class of problems which are in the difference of two NP classes of problems [Papadimitriou, 1994]. Wooldridge and Dunne [2004] have shown that similar decision problems for Qualitative Coalitional Games (such as minimal successful coalition) are D^p-complete.

We can now consider the following decision problems for the q-minimal-core of CAGs.

QUORUM MINIMAL CONSENSUS ACTION (QMCA)

Given a CAG $\Gamma = \langle G, A, J, q \rangle$, and a joint action $j \in J$, is j a q-minimal consensus action?

Algorithm: The algorithm must verify that $G' = agents(j)$ is the least fixed point of $Support_j$, and that no other quorum consensus action for G' reaches the least fixed point of $Support_j$ in fewer rounds than required for j.

Time Complexity: $O(n \times log(n) \times |J|)$

Algorithm 3 Is j a q-minimal consensus action.

```
function QMCA(j, Γ)
    G' ← agents(j)
    r ← 0
    if QCA(j, Γ) then
        r ← rounds(j)
    else
        return false
    for all k ∈ J do
        if agents(k) = G' ∧ QCA(k, Γ) then
            if rounds(k) < r then
                return false
    return true
```

A non-deterministic algorithm for deciding that j is *not* a quorum minimal consensus action will first check whether it is a consensus action (and return yes if it is not) and if it is, compute $rounds(j)$ and guess an action $k \in J$ and verify that $agents(k) = agents(j)$ and $rounds(k) < rounds(j)$. The problem of deciding that j is *not* a quorum minimal consensus action is therefore in NP(n) on RAM. Hence deciding whether j is a quorum minimal consensus action is in co-NP(n) on RAM.

QUORUM MINIMAL GROUP CONSENSUS (QMGC)

Given a CAG $\Gamma = \langle G, A, J, q \rangle$ and a subset of agents $G' \subseteq G$, is there a q-minimal consensus action for G'?

Algorithm: Observe that if G' has a quorum consensus action then G' has a q-minimal consensus action; therefore this problem is equivalent to QCG.

QUORUM MINIMAL NO CONSENSUS (QMNC)

Given a CAG $\Gamma = \langle G, A, J, q \rangle$, is it the case that no subset $G' \subseteq G$ has a q-minimal consensus action?

Algorithm: Observe that if any G' has a quorum consensus action then at least one G' has a q-minimal consensus action. Therefore this problem is equivalent to QNC.

A summary of our results is given in table 1.

	QC	G'-minimal	q-minimal
CA	$P(n)$	$co\text{-}NP(n)$	$co\text{-}NP(n)$
GC	$NP(n)$	$D^p(n)$	$NP(n)$
NC	$co\text{-}NP(n)$	$co\text{-}NP(n)$	$co\text{-}NP(n)$

Table 1: Summary of Results (upper bounds). QC – Quorum Consensus, CA – Action Consensus, GC – Group Consensus, NC – No Consensus. Note that we assume random access to indices in J, so the complexity classes are for (N)RAM.

4 Related Work

CAGs have some similarities to Qualitative Coalitional Games (QCGs) [Wooldridge and Dunne, 2004]). It is therefore interesting to compare CAGs and QCGs, especially with respect to the size of representation and the complexity of similar decision problems.

A QCG Γ may be represented as an $(n+3)$ tuple $\Gamma = \langle \mathsf{G}, \mathsf{Ag}, \mathsf{G_1} \ldots \mathsf{G_n}, \mathsf{V} \rangle$ where $\mathsf{G_i} \subseteq \mathsf{G}$ represents each agent's $\mathsf{i} \in \mathsf{Ag}$ set of goals and $\mathsf{V} : 2^{\mathsf{Ag}} \rightarrow 2^{2^{\mathsf{G}}}$ is the characteristic function of the game mapping each possible coalition of agents to the sets of goals that coalition can achieve. In QCGs:

- A set of goals $\mathsf{G}' \subseteq \mathsf{G}$ is *feasible* for a coalition $\mathsf{C} \subseteq \mathsf{Ag}$ if $\mathsf{G}' \in \mathsf{V}(\mathsf{C})$.
- A set of goals $\mathsf{G}' \subseteq \mathsf{G}$ *satisfies* an agent $\mathsf{i} \in \mathsf{C} \subseteq \mathsf{Ag}$ if $\mathsf{G}' \cap \mathsf{G_i} \neq \emptyset$.
- A coalition $\mathsf{C} \subseteq \mathsf{Ag}$ is *successful* if there exists some set of goals $\mathsf{G}' \subseteq \mathsf{G}$ such that G' is feasible for C and G' satisfies at least all agents $\mathsf{i} \in \mathsf{C}$. A coalition C is *selfishly successful* if G' is feasible for C and satisfies only the agents in $\mathsf{i} \in \mathsf{C}$.
- A coalition $\mathsf{C} \subseteq \mathsf{Ag}$ is in the *core* if it is both (selfishly) successful and minimal, i.e., there is no strict subset $\mathsf{C}' \subset \mathsf{C}$ which is successful.

To compare QCGs and CAGs, we can identify agents' goals with quorum consensus actions that they would participate in. CAGs thus correspond to a particular kind of QCGs, namely those where the characteristic function consists of singleton sets (since the agents can perform only one joint action at a time).

The worst case size of the game representation for QCGs is the characteristic function where each coalition can enforce any subset of goals. There are 2^n coalitions and 2^m subsets of goals, so the worst case size of V is $O(2^{n+m})$. This is different from CAGs where the worst case size of J is only exponential in n but not in m.

Complexity results for QCGs in [Wooldridge and Dunne, 2004] are given as a function of the size of representation, where the characterisitic function is replaced by a propositional formula Ψ (which as noted may be exponential in the number of agents and goals, but generally will be more concise than a naive representation of V). The successful coalition problem is NP in the size of the representation. It corresponds to our QGC problem which is also in NP, however it is NP in the number of agents (assuming random access). Alternatively, QGC can be characterised as linear in the size of representation since it involves a single iteration over J, doing a linear (in n) amount of work.

Relationships between CAGs and other game theoretic models can also be identified. A central premise in CAGs is that agents' choices are conditioned by the number of other agents also making some choice. Anonymous Games [Daskalakis and Papadimitriou, 2007] consider situations where the utility of participation in some coalition is independent of the identities of the agents concerned; in such situations other factors, including the size of the coalition become determinants of an agent's choice. In general, however, CAGs are non-anonymous therefore, for example, an agent could refuse ($q_i(j) = 1$) to participate in any joint action in which some other, specific, agent participates. In Imitation Games [McLennan and Tourky, 2010] two players take the roles of leader and follower; through the payoff structure the follower is motivated to act in consensus with the leader. McLennan and Tourky [2010] find that the complexity of several decision problems concerning Nash equilibria in such is games is no less than for the general two-player case.

CAGs are also related to work on conditional preference. In a CAG the agents must choose between potentially exponentially many joint actions. For an individual agent each joint action encodes: an action for that agent, the subset of agents with which it acts and the actions performed by those agents. Agents in CAGs must therefore make decisions over multiple domains.

It is not our intention that the quorum function be interpreted as a comparator or scale of preference over joint actions; however certain basic correspondences between the quorum function and preferences do exist. For example it is reasonable to identify those joint actions for which $q_i(j) = 0$ as being the 'most preferred' joint actions of agent i. Where $q_i(j) > 0$ support for a joint action becomes conditional, and an agent will only support j if the proportion of other agents supporting j exceeds $q_i(j)$.

Boutilier *et al* [1999] have proposed conditional preference, or CP-nets, as a natural and compact representation suitable for capturing conditional preferences over combinatorial domains. Succinctness is a useful property, as explicit representation of preference over exponentially many outcomes is often impractical. Preferences in CP-nets are formed under the assumption of ceteris paribus (all else being equal) and can be described as having the form: given x, $y > z$. This gives rise to preference structures which are potentially non-linear and may be incomplete.

There is considerable work in the social choice literature on preference aggregation. Much of this work has focused on the problem of aggregating the preferences of a large number of decision makers when making decisions over a single, relatively small, domain. Comparatively little attention has been given to collective decisions where the reverse is true, as is the case for CAGs. A notable exception is [Lang, 2007] where the potential of structure within CP-nets to reduce the computational overhead associated with combinatorial problems is explored. However Lang [2007] has shown that the complexity of all positional scoring voting rules, including Borda and even simple majority, cannot be reduced using CP-nets.

5 Discussion and Future Work

We have introduced consensus action games, in which agents' willingness to participate in joint actions is mediated by a biologically inspired quorum function. We have analysed the complexity of several natural decision problems associated with individual and collective rationality in CAGs and shown that tractable algorithms exist (at worst polynomial in the size of the input). We conjecture that the upper bounds are tight (that the lower bounds for the problems in table 1 are the same).

We have chosen to study consensus action selection in a context where individual decisions are conditioned through a quorum response as opposed to the more common setting where decisions are guided by preference. It seems likely that collective decisions in natural systems are not taken on a purely preferential basis; inherent difficulties associated with the representation, elicitation and aggregation of preferences in combinatorial domains are well known. Our results suggest that quorum behaviours may make comparatively lower cognitive demands on a decision maker. This may explain why even the simplest organisms are able to effectively coordinate group-level activities through the quorum mechanism.

A robust decision making procedure should reliably produce beneficial outcomes for the decision makers under diverse conditions. Our present model considers agents acting under a single set of constraints, those joint actions given in J. A natural extension to this work would be to consider iterated consensus action selection, where decisions regarding joint actions are taken repeatedly in differing states of the world. An iterated version of CAGs would allow us to investigate the performance of quorum consensus decisions over time. For example, it would be interesting to examine the implications of this decision process for both individual and social welfare. Of equal interest are the questions of how an agent's quorum function is implemented, the strategies that agents may employ in selecting quorum thresholds and the effects of these on individual and group well being.

References

[Boutilier *et al.*, 1999] C. Boutilier, R. I. Brafman, H. H. Hoos, and D. Poole. Reasoning with conditional ceteris paribus preference statements. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 71–80. Citeseer, 1999.

[Conradt and Roper, 2005] L. Conradt and T. J. Roper. Consensus decision making in animals. *Trends in Ecology and Evolution*, 20:449–456, 2005.

[Daskalakis and Papadimitriou, 2007] C. Daskalakis and C. Papadimitriou. Computing equilibria in anonymous games. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 83–93. IEEE, 2007.

[Grosz and Kraus, 1993] Barbara Grosz and Sarit Kraus. Collaborative plans for group activities. In *IJCAI'93: Proceedings of the 13th international joint conference on Artifical intelligence*, pages 367–373, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[Immerman, 1998] Neil Immerman. *Descriptive complexity*. Springer, 1998.

[Jacob *et al.*, 2004] E.B. Jacob, I. Becker, Y. Shapira, and H. Levine. Bacterial linguistic communication and social intelligence. *TRENDS in Microbiology*, 12(8):366–372, 2004.

[Lang, 2007] J. Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1366–1371, 2007.

[Levesque *et al.*, 1990] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 94–99. Boston, MA, 1990.

[List, 2004] Christian List. Democracy in animal groups: a political science perspective. *Trends in Ecology & Evolution*, 19(4):168–169, April 2004.

[McLennan and Tourky, 2010] Andrew McLennan and Rabee Tourky. Simple complexity from imitation games. *Games and Economic Behavior*, 68(2):683 – 688, 2010.

[Papadimitriou, 1994] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[Pratt and Sumpter, 2006] S. C. Pratt and D. J. T. Sumpter. A tunable algorithm for collective decision-making. *Proceedings of the National Academy of Sciences*, 103(43):15906, 2006.

[Pratt *et al.*, 2005] S. C. Pratt, D. J. T. Sumpter, E. B. Mallon, and N. R. Franks. An agent-based model of collective nest choice by the ant temnothorax albipennis. *Animal Behaviour*, 70(5):1023–1036, 2005.

[Seeley and Visscher, 2004] T.D. Seeley and P.K. Visscher. Group decision making in nest-site selection by honey bees. *Apidologie*, 35(2):101–116, 2004.

[Stueckle and Zinner, 2008] Sabine Stueckle and Dietmar Zinner. To follow or not to follow: decision making and leadership during the morning departure in chacma baboons. *Animal Behaviour*, 75(6):1995–2004, June 2008.

[Sumpter and Pratt, 2008] D. J. Sumpter and S. C. Pratt. Quorum responses and consensus decision making. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 2008.

[Tambe, 1997] Milind Tambe. Towards flexible teamwork. *CoRR*, cs.AI/9709101, 1997.

[Ward *et al.*, 2008] A. J. W. Ward, D. J. T. Sumpter, I. D. Couzin, P. J. B. Hart, and J. Krause. From the cover: Quorum decision-making facilitates information transfer in fish shoals. *Proceedings of the National Academy of Sciences*, 105(19):6948, 2008.

[Wooldridge and Dunne, 2004] Michael Wooldridge and Paul E. Dunne. On the computational complexity of qualitative coalitional games. *Artif. Intell.*, 158(1):27–73, 2004.