# An Essay on Sabotage and Obstruction

Johan van Benthem

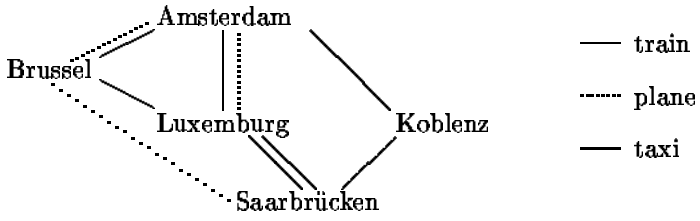
ILLC Amsterdam & CSLI Stanford

To Jörg Siekmann on the occasion of his $60^{th}$ Birthday

**Abstract.** This is a light 'divertissement' about the problems of modern transportation, and the beckoning pleasures of meeting with Jörg.

### Getting nowhere

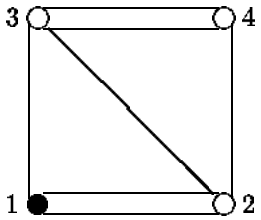
Traveling between Saarbrücken and Amsterdam is easy, as Jörg well knows. A convenient network of connections exists between these two capitals of logic and computation. In the phantasy world of this essay, the network is as pictured below. But what if the transportation system breaks down, and a malevolent demon starts canceling connections, anywhere in the network? Of course, a veteran AI researcher adapts, and changes to the next optimal plan. But what if, at every stage of his trip, the demon first takes out one connection? From Saarbrücken to Amsterdam, Jörg still has a winning strategy. The Demon's opening move may block Brussel or Koblenz, but then Jörg goes to Luxemburg in the first round, and gets to Amsterdam in the next. The Demon may also cut a connection between Amsterdam and some city in the middle - but Jörg can then go to at least one place offering him still two intact roads. My own Dutch situation is less rosy, however. This time, Demon has the winning strategy. It starts by cutting a link between Saarbrücken and Luxemburg. If I now go to any city in the middle, Demon always has time in the next rounds to cut my beckoning link to Saarbrücken. Oh, that fair city on the Saar, and yet so inaccessible!




This story may sound like a bad fairy tale - but users of the Dutch Railway System NS will recognize the prevalent situation since one year. Connections disappear in malevolent patterns, and what used to be simple travel has become a game of high complexity. This essay reflects the thoughts of a traveling logician stuck at strange stations...

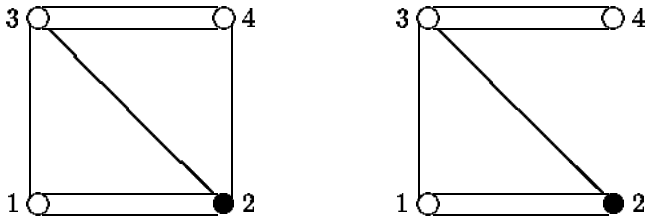
**From algorithms to obstruction games**

The preceding example is the well-known task of *Graph Reachability* 'sabotaged'. More generally, any algorithmic task over graphs can be turned into a two-player game, with one player 'Runner' trying to do the original job, and another player 'Blocker' taking out edges at each stage. Different schedulings and winning conditions are possible. For instance, consider a game which sabotages *Traveling Salesman*. An undirected graph is given, and Runner must complete a circuit. This time, Blocker lets him start in each round, and then takes out a link. Players must move as long as they can: the game stops the first time a player cannot move. Runner wins if the end situation contains a completed circuit; otherwise, Blocker wins. Which player has a winning strategy in the following game, which starts with Runner at the black dot?




It may take a moment: but then you will see that Blocker has the winning strategy, first cutting one upper link. (Complex games of this sort are a nice pastime for winter evenings.) But, why this implicit trust that one of the two players must have a winning strategy - i.e., that the game is *determined*? Recall *Zermelo's Theorem* from the dawn of game theory. It says that each finite two-player zero-sum game is determined. Of course, the great German set theorist was thinking about Chess when he proved his result - as the Reichsbahn was still running according to schedule in his days - but the mathematics applies equally well to our modern transport plight. The reason is that the sabotage games still satisfy Zermelo's three conditions.

One can see this as a *game transformation*. The original algorithmic task is a graph game with just one player, where that player must perform a sequence of moves creating a path with some desired property ('ending in a specified location', 'forming a circuit', etc.). Now we get a new game, whose local states are subtrees of the original game tree, with a current position for Runner indicated, whose moves are of two kinds. Runner can follow an edge from his current position in the given graph, but Blocker can choose a new graph missing one edge. For instance, the Traveling Salesman game has the given diagram as an initial node. We display one opening move for Runner, followed by one for Blocker:



This blow up of the original search space to a game tree involves an exponential factor - though Blocker's moves also simplify the game as the graph gets simpler.

**Complexity bounds and model checking**

Intuitively, solving my travel problems, if possible, against sabotage seems more complex than the old task itself. Graph Reachability is in **P**, Traveling Salesman is **NP**-complete. We will not determine the exact complexity of their sabotaged versions, but some quick general observations can be made. First, note that both problems amount to *model-checking* with *first-order* formulas. Given any finite graph **G**, they state the existence of a sequence of points satisfying some first-order definable condition, i.e., both check an *existential* formula of the form:

$$\exists x_1 \ldots \exists x_k \;\; \alpha(x_1, \ldots, x_k) \qquad \text{with } \alpha \text{ quantifier-free}$$

where $k$ is of the order of the size of **G**. More conveniently, think of the graph as a domain of edges, and let the existential quantifiers run over these. What is the effect of Blocker's activities? At each stage, one edge is removed arbitrarily. The result of this progressive impoverishment is that Runner has a winning strategy if and only if the following first-order formula over edges is true in the graph **G**:

$$\exists x_1 \forall y_1 \exists x_2 \neq y_1 \; \forall y_2 \exists x_3 \neq y_1, y_2 \;\; \ldots \;\; \alpha'(x_1, \ldots, x_k)$$

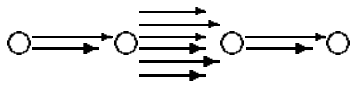
where $\alpha'$ is the old condition suitably adapted in terms of endpoints of edges. The length of this second formula is still linear in the size of the graph, but we have quantifier interchanges now. Thus, an upper bound for the complexity is that of uniform model checking of first-order formulas over finite models, which takes polynomial space in the size of the model plus that of the formula.

**Fact** Solving a sabotage game takes at most **PSPACE** in the graph size.

Probably, one cannot do better in general. Here is one lower bound. Consider the **PSPACE**-complete problem of *Quantified Boolean Formulas*. This is akin to a sabotaged Reachability task. Consider this special case:

$$\forall p_1 \exists p_2 \forall p_3 \;\; \alpha(p_1, p_2, p_3) \qquad \text{with } \alpha \text{ some propositional formula}$$

Now look at a graph with 3 nodes, and edges distributed as follows:



thick edges : label *true*
thin edges : label *false*

Here is a sabotage game over this graph. Blocker starts each round by taking away an edge, after which Runner chooses an edge to cross. The game ends when Runner can no longer move - and such a situation is counted as a win for Blocker `iff` it is an end point and the formula $\alpha(p_1, p_2, p_3)$ evaluated according to the labels *true* or *false* of the successive edges chosen by Runner is false. Now, let us analyze the strategic situation in this game. Blocker must let Runner move three times: otherwise he loses. But by taking away links, he can force the truth value at the first step, and then at the third - or perhaps just at the third. For Runner to succesfully counter all possible actions by Blocker requires the truth of the following formulas:

$\forall p_1 \exists p_2 \forall p_3 \; \alpha(p_1, p_2, p_3)$
$\forall p_1 \forall p_3 \exists p_2 \; \alpha(p_1, p_2, p_3)$
$\forall p_3 \exists p_1 \exists p_2 \; \alpha(p_1, p_2, p_3)$
$\exists p_1 \exists p_2 \forall p_3 \; \alpha(p_1, p_2, p_3)$
$\exists p_1 \exists p_2 \exists p_3 \; \alpha(p_1, p_2, p_3)$

But the second to fifth formulas are implied by the first. Thus, Runner has a winning strategy in this game iff the first quantified Boolean formula is true. I suspect that the general situation with $k$ propositional variables yields to a similar construction, which suggests that

Sabotaged graph tasks can be **PSPACE**-hard games.

Admittedly, the preceding example is contrived - and its trick does not work for the above sabotaged Reachability or Traveling Salesman. On the other hand, John Bell pointed at the related (though more involved) game 'Roadblock' in David Harel's beautiful book *Algorithmics*, which takes even essentially **EXPTIME**. The general complexity behaviour of sabotage still eludes me.

**Modal logics over changing models**

Instead of solving our complexity problem, we will look at it from another angle. Solving, say, a standard reachability problem amounts to evaluating a modal formula with a number of disjunctions of the form $\Diamond \ldots \Diamond p$ , where $p$ holds in the goal states. Uniform model checking for modal formulas on a finite model is known to be in **P**-time - measured in the size of the model and the length of the formula. But the above sabotaged versions involve *changing the model* as we proceed. Here is a way of viewing this, thinking of models where arrows are treated as objects. Introduce a cross-model modality referring to submodels from which objects have been removed:

$\mathbf{M}, s \models \Diamond\!\!\!-\phi$ `iff` there is a world $w \neq s$ with $\mathbf{M}\text{-}\{w\}, s \models \phi$

Now the language has both an 'internal modality' $\Diamond$ and an 'external modality' $\Diamond\!\!\!-$, which can be combined. E.g., the fact that universal modal formulas

are preserved under submodels shows in valid principles like $\Box p \rightarrow \boxminus\Box p$. Actually, despite the modal notation, this language lacks some typical modal features.

For instance,

the formula $\blacklozenge\Box \bot$ is not invariant for bisimulations.

This formula holds in an irreflexive 2-cycle, but it fails in the bisimilar model consisting of a single reflexive point. Nevertheless, the formulas of this language are still translateable into an obvious first-order language, using the same trick as above. E.g., $\blacklozenge\Box \bot$
says that $\exists y \neq x \, \neg\exists z \neq y : Rxz$.

The blow-up in this translation is only polynomial, and hence model checking in the new language can be performed in **PSPACE**, the complexity of uniform model checking for first-order formulas. But is this upper bound also a lower one? After all, the model shrinks when we make a jump via the operator $\blacklozenge$. The same open question of complexity now returns:

*What is the complexity of uniform model checking for the $\Diamond$, $\blacklozenge$ language?*

Another obvious open questions concerns the complete logic of these operators. Logics like this axiomatize a bit of the meta-theory of modal evaluation plus some natural model operations. One could also add modalities for passing to arbitrary submodels (on finite models, this is the Kleene iteration of $\blacklozenge$), and other model constructions have been considered as well. E.g., in modern update logics for communication, a public announcement A! of an assertion A restricts the current model to only those worlds where A holds. Evaluation of further modal statements about agents' knowledge and ignorance then moves to a *definable submodel*, driven by mixed assertions such as $\Box_{A!}\mathrm{K}_i\phi$. And there are still further examples in the literature. The above is just a start. New questions of model checking and completeness emerge in all such systems.

### Disturbing a game in general

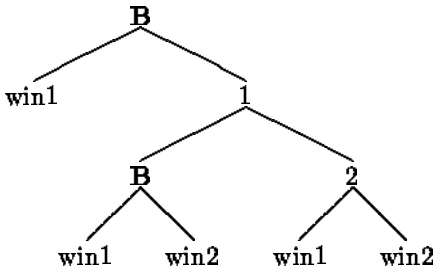
Time for some free association! We turned an algorithm into a game by introducing a disturbing player modeling the action of a hostile environment. But we can also disturb *any game* itself. One simple variant is as follows. Consider a game tree, with one player for convenience - and at the start of each round, let Blocker *prune one game move* from the tree. The effects of this local disturbance are not so dramatic. One can still compute winning positions for Runner in the original game tree, in the same inductive fashion as with Zermelo's algorithm for finding the winning player in finite game trees. The key observation is this:

**Fact** At any game node, Runner has a winning strategy against Blocker `iff` he has winning strategies in the subtrees for *at least two* of his moves.

From right to left, this is obvious, as Blocker can only affect at most one of those subtrees in the first round, leaving Runner free to go to the other one. From left to right, if there was at most one such subtree, then Blocker can cut the move to that, and force Runner to choose a move to a losing position. The resulting inductive algorithm for computing winning positions is about as simple as that of Zermelo. But, our sabotaged graph tasks are not like this! Blocker removed an edge from the graph, which cuts travel options for Runner at many stages in the original search space. This is a global action, which amounts to *pruning a whole set of moves simultaneously* from a game tree. The effects of this are not as easy to compute inductively - again reflecting the essentially greater complexity of the new game.

**Social life and coalition logic**

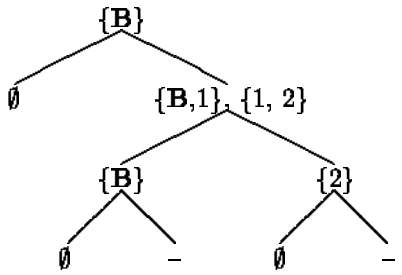
Sabotaging players also make sense in $n$-person games, whether removing single moves or whole sets of them. The corresponding game transformation produces an $(n+1)$-person game, and the above issue of winning or losing generalizes to *coalitional powers* of groups of players, including Blocker. Here is an example:




Working upward, inductively, one can compute *forcing coalitions* for any proposition $p$. At bottom nodes satisfying $p$, these are the empty set of players $\emptyset$ and its supersets. At end nodes not satisfying $p$, no forcing coalition exists for $p$. At higher nodes which are turns for player $j$ , the rule is this:

Take all $p$-forcing coalitions that occur at daughters and add $j$ to them.

To simplify the calculation, *supersets may be suppressed* here - as these represent weaker derived powers of groups. For the above game tree, with $p$ the predicate win1, the minimal $p$-forcing coalitions would be:




But the logic of sabotage supports more refined new notions, such as Blocker's being able to determine the outcome of the game - in the sense of being able to make either player 1 or player 2 win. Also, players can make a pact with the Devil, and oppose their old antagonists more effectively. Modern modal game logics describe these complex statements of powers for individuals and coalitions in a general way. With our earlier games on graphs **G** , such statements still translate into first-order properties of **G** - and the earlier observation about model checking still applies.

Thus, after the game transformation, many interesting aspects of sabotage become matters of coalitional game theory - and with graph games, even of standard first-order logic.

**Receptions, circulation, and pacts with the devil**

This mathematical calculus is closer to real life than you might think. Directors of institutes like Jörg need the skill of *circulation at receptions*. Suppose there are 5 positions, with your starting point at the black dot, and mine at the grey one. At each round, we have to shift position: you move first, and I follow. One important skill among academics of high standing is "meet" versus "avoid". Here is a picture of an initial situation plus the situation after one possible round:

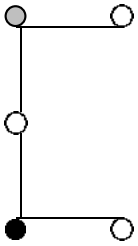
Analysing the strategic situation, one easily finds the following three 'powers':

(a) You can force us to meet,
(b) I can force us to meet,
and therefore, none of us can force 'avoidance' - but still:
(c) You and I *together* can force us to avoid each other,
most easily by cycling back and forth at two disjoint links.

More spectacular avoidance strategies for one player against another occur around cycles, much like sequences of running around obstacles in action movies. Now introduce a Blocker who can take away links, say a manipulative host. The new power structure will depend on the scheduling. Consider first a rule where Blocker starts each round, then you, then me. Then it is easy to see that

Blocker can force us to meet, but also force avoidance.

He forces a meet by cutting links to decrease our room of manoevre, but still in one connected component. He forces us to avoid each other by imprisoning us inside disjoint components. Pacts with the devil make no sense here, as Blocker controls all relevant outcomes anyway. With a rule: "First you, then me, then Blocker", you retain your earlier power to force a meet, Blocker can force the same, and I have no significant powers. A pact with the devil does make sense at the following reception:

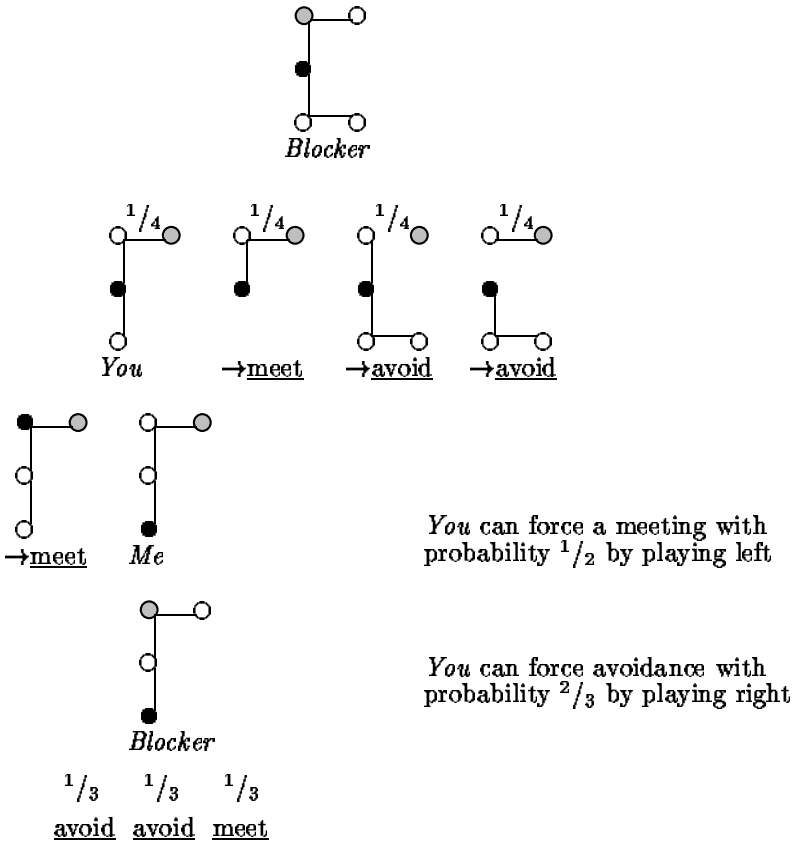


Neither you nor I alone have the power to force a meet here, or force avoidance. But the *coalition* {you, me} can force a meet, and it can also force avoidance - by both going to our nearest end-points: Blocker must then cut us off in the middle. Also, in a pact with Blocker, each of us can force a meet, or avoidance.

These outcomes show that the above general Meet and Avoid strategies for Blocker alone have their limitations. Still, it is easy to see that they will always work if he has *empty moves*, allowing him to wait until we have made our compulsory moves. Another aspect of social algorithmics is potential redesign. Once powers for meeting and avoiding have been computed for a reception scheme, a host might want to design simpler - perhaps cheaper - social events with the same effect. Moreover, these are not mere formal phantasies. When the German president von Weiszäcker visited Groningen University in the early 1980s, we professors at his gala reception were all given a mathematical circulation diagram with exact choreographic instructions on how to move from table to table. I guess the much greater professionalism at our modern Dutch universities also includes instructions on what to *say*...

**The meaning of sabotage**

Any algorithmic task can be 'sabotaged' to create real-life versions in a hostile environment. This can be done by turning it into a game, which can be studied by known techniques. Thus, in Clausewitz' immortal phrasing, game theory is "algorithmics pursued by other means". But there are also other perspectives on these phenomena. In particular, the logical model-checking angle suggests a study of evaluation of first-order logic on structures *which change under evaluation*. E.g., an object might become unavailable once drawn from the domain, or a fact might change when inspected (think of measurement in quantum mechanics). This would be like adding aspects of sabotage to logical evaluation games. Finally, in the realities of modern transport, sabotage comes in a more global and *statistical* fashion. Thus, we might also assume that Blocker is a blind opponent cutting links with equal probability. This will generally improve the situation for Runner, and one can use a Zermelo algorithm to compute his *expected game values*, indicating the degree of deterioration from Runner's prospects in the initial task. E.g., with a random Blocker, here is a piece of the game tree for the second reception, whose schedule was ' *You, Me, Blocker*':



This is more like known tasks in Graph Theory, where *random graph problems* may involve removal of nodes and edges. By contrast, our opponents in this essay are maximally malevolent. More positively, canceled connections may come *alive* again in the course of our trip when we add a third player 'Deblocker'. Lots of further complications to investigate! But then, the field is still young.

Even so, one deeper question remains. *Why* does the Dutch railway system behave in its current erratic mode? Instead of thinking negatively about failure and doom here, we can also think positively about the intentions of our political leaders who govern it. For many centuries, the average member of our nation has muddled through life in **P**-like, or at best **NP**-like patterns of behaviour. But now, in the new millennium, the authorities have decided that we have reached a new plateau of intelligence. The Dutch are ripe for **PSPACE** tasks, and we are given a challenging chance to practice these. Similar encouraging experiences are reported from the British Isles. This represents a new stage in human evolution, and I am sure we will see many similar phenomena all across Europe soon!

**Acknowledgment**
Peter van Emde Boas and Hanno Hildmann provided indispensable support.

**Postscript**
In the few months between the first posting of this essay and publication, Christof Löding and Philipp Rohde (RWTH Aachen) have answered the main complexity question. Through an ingenious construction, sabotaged Graph Reachability and Traveling Salesman both turn out PSPACE-hard on finite graphs.