Manipulating Boolean Games Through Communication

John Grant[1,2] **Sarit Kraus**[2,3] **Michael Wooldridge**[4] **Inon Zuckerman**[2]

[1]Department of Mathematics,
Towson University, Towson, D 21252, USA

[2]Institute for Advanced Computer Studies
University of Maryland, College Park 20742, USA

[3]Department of Computer Science
Bar-Ilan University, Ramat-Gan, 52900 Israel

[4]Department of Computer Science,
University of Liverpool, Liverpool L69 3BX, UK

Abstract

We address the issue of manipulating games through communication. In the specific setting we consider (a variation of Boolean games), we assume there is some set of environment variables, the value of which is not directly accessible to players; each player has their own beliefs about these variables, and makes decisions about what actions to perform based on these beliefs. The communication we consider takes the form of (truthful) announcements about the value of some environment variables; the effect of an announcement about some variable is to modify the beliefs of the players who hear the announcement so that they accurately reflect the value of the announced variables. By choosing announcements appropriately, it is possible to perturb the game away from certain rational outcomes and towards others. We specifically focus on the issue of *stabilisation*: making announcements that transform a game from having no stable states to one that has stable configurations.

1 Introduction

Our aim in the present paper is to investigate the use of communication in the management and control of multi-agent systems. In particular, we look at how announcements that affect the beliefs of players in a game can be used to *stabilise* the game. The games we consider are a variant of Boolean games [11; 3; 6; 7]. In Boolean games, each player in the game has under its unique control a set of Boolean variables, and is at liberty to assign values to these variables as it chooses. In addition, each player has a goal that it desires to be achieved: the goal is represented as a Boolean formula, which may contain variables under the control of other players. The fact that the achievement of one agent's goal may depend on the choices of another agent is what gives Boolean games their strategic character. In the variant of Boolean games that we consider in the present paper, we assume that each player has (possibly incorrect) beliefs about a certain set of *environment variables*, which have a fixed value, outside the control of any players in the game. An external *principal* is assumed to be able to (truthfully) announce the value of (some subset of) environment variables to the players in the game. Announcing a variable has the effect of changing the beliefs of players in the game, and hence, potentially, their preferences over possible outcomes. By choosing announcements appropriately, the principal can perturb the game away from some possible outcomes and towards others.

We focus particularly on the issue of *stabilisation*: making announcements that transform a game from having no stable states to one that has stable configurations. Stability in this sense is close to the notion of Nash equilibrium in the game-theoretic sense [14]: it means that no agent has any incentive to unilaterally change its choice. However, the difference between our setting and the conventional notion of Nash equilibrium is that an agent's perception of the utility it would obtain from an outcome is dependent on its own beliefs. By changing these beliefs through truthful announcements, we can modify the rational outcomes of the game. We are particularly interested in the possibility of transforming a game that has no stable states into one that has at least one. Our rationale for this consideration is that instability will, in general, be undesirable: apart from anything else, it makes behaviour harder to predict and understand, and introduces the possibility of players wasting effort by continually modifying their behaviour. It makes sense, therefore, to consider the problem of stabilising multi-agent system behaviour: of modifying an unstable system so that it has equilibrium states, and even further, of modifying the system so that it has socially desirable equilibria. For example, we might consider the principal perturbing a game to ensure an equilibrium that maximises the number of individual agent goals achieved.

Although the model of communication and rational action we consider in the present paper is based on the abstract setting of Boolean games, the issues we investigate using this model – stabilisation, and, more generally, the management of multi-agent systems – are, we believe, of central importance. This is because there is a fundamental difference between a distributed system in which all components are designed and implemented by a single designer, and which can therefore be designed to act in the furtherance of the designer's objectives, and multi-agent systems, in which individual agents will selfishly pursue their own goals. By providing a formal analysis of how communication can be used to perturb the rational actions of agents within a system towards certain outcomes, we provide a foundation upon which future, richer models can be built and investigated.

2 The Model

In this section, we introduce the model of Boolean games that we work with throughout the remainder of this paper. This model is a variation of previous models of Boolean games [11; 3; 6; 7]. The main difference is that we assume players in the game have *beliefs* about a set of *environment variables*, the values of which are not under the control of any players in the game. These beliefs may be incorrect. Players base their decisions about what choices to make based on their beliefs.

Propositional Logic: Let $\mathbb{B} = \{\top, \bot\}$ be the set of Boolean truth values, with "$\top$" being truth and "$\bot$" being falsity. We will abuse notation a little by using $\top$ and $\bot$ to denote both the syntactic constants for truth and falsity respectively, as well as their semantic counterparts. Let $\Phi = \{p, q, \ldots\}$ be a (finite, fixed, non-empty) vocabulary of Boolean variables, and let $\mathcal{L}$ denote the set of (well-formed) formulae of propositional logic over Φ, constructed using the conventional Boolean operators ("$\wedge$", "$\vee$", "$\rightarrow$", "$\leftrightarrow$", and "$\neg$"), as well as the truth constants "$\top$" and "$\bot$". Where $\varphi \in \mathcal{L}$, we let $vars(\varphi)$ denote the (possibly empty) set of Boolean variables occurring in φ (e.g., $vars(p \wedge q) = \{p, q\}$). A *valuation* is a total function $v : \Phi \rightarrow \mathbb{B}$, assigning truth or falsity to every Boolean variable. We write $v \models \varphi$ to mean that the propositional formula φ is true under, or satisfied by, valuation v, where the satisfaction relation "$\models$" is defined in the standard way. Let $\mathcal{V}$ denote the set of all valuations over Φ. We write $\models \varphi$ to mean that φ is a tautology. We denote the fact that $\models \varphi \leftrightarrow \psi$ by $\varphi \equiv \psi$.

Agents and Variables: The games we consider are populated by a set $Ag = \{1, \ldots, n\}$ of *agents* – the players of the game. Each agent is assumed to have a *goal*, characterised by an $\mathcal{L}$-formula: we write γ_i to denote the goal of agent $i \in Ag$. Agents $i \in Ag$ each *control* a (possibly empty) subset Φ_i of the overall set of Boolean variables. By "control", we mean that i has the unique ability within the game to set the value (either $\top$ or $\bot$) of each variable $p \in \Phi_i$. We will require that $\Phi_i \cap \Phi_j = \emptyset$ for $i \neq j$, but in contrast with other existing models of Boolean games [11; 3], we do *not* require that $\Phi_1, \ldots, \Phi_n$ form a partition of Φ. Thus, we allow for the possibility that some variables are not under the control of any players in the game. Let $\Phi_E = \Phi \setminus (\Phi_1 \cup \cdots \cup \Phi_n)$ be the variables that are not under any agent's control; we call these the *environment* variables. It is assumed that the value of these variables is determined in some way external to the game, and that agents within the game cannot influence these variables in any way: the actual value of the variables Φ_E is immutable. We let $v_E : \Phi_E \rightarrow \mathbb{B}$ be a function that gives the actual value of the environment variables. When playing a Boolean game, the primary aim of an agent i will be to choose an assignment of values for the variables Φ_i under its control so as to satisfy its goal γ_i. The difficulty is that γ_i may contain variables controlled by other agents $j \neq i$, who will also be trying to choose values for their variables Φ_j so as to get their goals satisfied; and their goals in turn may be dependent on the variables Φ_i. In addition, goal formulae may contain environment variables Φ_E, beyond the control of any of the agents in the system. A *choice* for agent $i \in Ag$ is a function $v_i : \Phi_i \rightarrow \mathbb{B}$, i.e., an allocation of truth or falsity to all the variables under i's control. Let $\mathcal{V}_i$ denote the set of choices for agent i.

Beliefs: In our games, we assume that agents have possibly incorrect beliefs about the value of environment variables Φ_E. We model the beliefs of an agent $i \in Ag$ via a function $\beta_i : \Phi_E \rightarrow \mathbb{B}$, with the intended interpretation that if $\beta_i(p) = b$, then this means that agent i believes variable $p \in \Phi_E$ has value b. It goes without saying that this is a simple model of belief, and that many alternative richer models of belief could be used instead: we could model an agent's beliefs as a set of valuations, yielding something like the possible worlds model for belief/knowledge, or we could model beliefs through a probability distribution over possible values for variables (see, e.g., [10]). However, modelling belief via functions $\beta_i : \Phi_E \rightarrow \mathbb{B}$ is conveniently simple, while at the same time allowing us to consider complex issues, as we will see later. We comment on this issue further in section 6.

Outcomes: An *outcome* is a collection of choices, one for each agent. Formally, an outcome is a tuple $(v_1, \ldots, v_n) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_n$. Notice that an outcome defines a value for all variables apart from the environment variables Φ_E: when taken together with a valuation v_E for environment variables, an outcome uniquely defines an overall valuation for the variables Φ, and we will often think of outcomes as valuations, for example writing $(v_1, \ldots, v_n, v_E) \models \varphi$ to mean that the valuation defined by the outcome $(v_1, \ldots, v_n)$ taken together with v_E satisfies formula $\varphi \in \mathcal{L}$. Notice that since v_E is assumed to be fixed for a game, we sometimes suppress reference to it when context makes the v_E function unambiguous, simply writing $(v_1, \ldots, v_n) \models \varphi$. A belief function β_i together with an outcome $(v_1, \ldots, v_n)$ also defines a unique valuation for Φ, and we will sometimes write $(v_1, \ldots, v_n, \beta_i)$ to mean the valuation obtained from the choices $v_1, \ldots, v_n$ together with the values for variables Φ_E defined by β_i. Observe that we could have $(v_1, \ldots, v_n, \beta_i) \models \gamma_i$ (intuitively, agent i believes it would get its goal γ_i achieved by outcome $(v_1, \ldots, v_n)$) while $(v_1, \ldots, v_n, v_E) \not\models \gamma_i$ (in fact i would not get its goal achieved by outcome $(v_1, \ldots, v_n)$). Let $succ(v_1, \ldots, v_n, v_E)$ denote the set of agents who have their goal achieved by outcome $(v_1, \ldots, v_n)$, i.e., $succ(v_1, \ldots, v_n, v_E) = \{i \in Ag \mid (v_1, \ldots, v_n, v_E) \models \gamma_i\}$.

Costs: Intuitively, the actions available to agents correspond to setting variables true or false. We assume that these actions have *costs*, defined by a *cost function* $c : \Phi \times \mathbb{B} \rightarrow \mathbb{R}_{\geq}$, so that $c(p, b)$ is the marginal cost of assigning variable $p \in \Phi$ the value $b \in \mathbb{B}$ (where $\mathbb{R}_{\geq} = \{x \in \mathbb{R} \mid x \geq 0\}$). Note that if an agent has multiple ways of getting its goal achieved, then it will prefer to choose one that minimises costs; and if an agent cannot get its goal achieved, then it simply chooses to minimise costs. However, cost reduction is a *secondary* consideration: an agent's primary concern is goal achievement.

Boolean Games: A Boolean game, G, is a $(3n + 4)$-tuple:

$$G = \langle Ag, \Phi, \underbrace{\Phi_1, \ldots, \Phi_n}_{\substack{\text{controlled}\\ \text{variables}}} \underbrace{\gamma_1, \ldots, \gamma_n}_{\text{goals}}, \underbrace{\beta_1, \ldots, \beta_n}_{\text{beliefs}}, c, v_E \rangle,$$

where $Ag = \{1, \ldots, n\}$ is a set of agents, $\Phi = \{p, q, \ldots\}$ is a finite set of Boolean variables, $\Phi_i \subseteq \Phi$ is the set of Boolean variables under the unique control of $i \in Ag$; $\gamma_i \in \mathcal{L}$ is the goal of agent $i \in Ag$; $\beta_i : \Phi_E \to \mathbb{B}$ is the belief function of agent $i \in Ag$; $c : \Phi \times \mathbb{B} \to \mathbb{R}_{\geq}$ is a cost function; and $v_E : \Phi_E \to \mathbb{B}$ is the (fixed) valuation function for environment variables.

Subjective Utility: We now introduce a model of utility for our games. While we find it convenient to define numeric utilities, it should be clearly understood that utility is not assumed to be transferable: it is simply a numeric way of capturing an agent's preferences. The basic idea is that an agent will strictly prefer all outcomes in which it gets its goal achieved over all outcomes where it does not; and secondarily, will prefer to minimise costs. Utility functions as we define them directly capture such preferences.

Of course, agents in our model have beliefs, modelled by belief functions β_i, and the choices agent i makes will depend on its beliefs β_i. When an agent makes a choice, it intuitively makes a calculation about the benefit it will obtain that takes into account its beliefs. However, this calculation is *subjective*, in the sense that the agent's beliefs may be wrong, and hence its judgement about the utility it will obtain from making a choice may be wrong. We let $u_i(v_1, \ldots, v_n)$ denote the utility that agent i believes it would obtain if agent j $(1 \leq j \leq n)$ made choice v_j. Formally, this value is defined as follows. First, with a slight abuse of notation, we extend cost functions to agents and individual choices. We let $c_i(v_i)$ denote the cost to agent i of choice v_i:

$$c_i(v_i) = \sum_{p \in \Phi_i} c(p, v_i(p))$$

Next, let v_i^e denote a choice for i that has the highest possible cost for i, and μ_i the cost to i of such a choice, formally:

$$v_i^e \in \arg\max_{v_i \in \mathcal{V}_i} c_i(v_i) \qquad\qquad \mu_i = c_i(v_i^e).$$

We then define the subjective utility that i would obtain from choices $v_1, \ldots, v_n$ by:

$$u_i(v_1, \ldots, v_n) = \begin{cases} 1 + \mu_i - c_i(v_i) & \text{if } (v_1, \ldots, v_n, \beta_i) \models \gamma_i \\ -c_i(v_i) & \text{otherwise.} \end{cases}$$

It is important to note that in this definition the value of an agent's utility is critically dependent on its beliefs β_i.

Nash Stability: We now define the notion of equilibrium that we use throughout the remainder of this paper: this notion of *Nash stability* is a variation of that defined in [9]. The basic idea of Nash stability, as with (pure strategy) Nash equilibrium [14], is that an outcome is said to be Nash stable if no agent within it would prefer to make a different choice, assuming every other agent stays with their choice. However, the difference between Nash stability and the conventional notion of Nash equilibrium is that an agent i in our setting will compute its utility – and hence make its choice – based on its beliefs β_i. We say an outcome $v_1, \ldots, v_i, \ldots, v_n$ is *individually stable for agent i* if $\nexists v_i' \in \mathcal{V}_i$ such that $u_i(v_1, \ldots, v_i', \ldots, v_n) > u_i(v_1, \ldots, v_i, \ldots, v_n)$. We then say an outcome $v_1, \ldots, v_n$ is Nash stable if $v_1, \ldots, v_n$ is individually stable $\forall i \in Ag$. We denote the Nash stable outcomes of a game G by $\mathcal{N}(G)$. As with Nash equilibrium, it may be that $\mathcal{N}(G) = \emptyset$; in this case we say G is *unstable*.

Example 1 *Consider the following (tongue-in-cheek) example. Bob likes Alice, and he believes Alice likes him. Although Bob doesn't like going to the pub usually, he would want to be there if Alice likes him and Alice was there also. Alice likes going to the pub, but in fact she doesn't like Bob: she wants to go to the pub only if Bob isn't there. We formalise this example in our setting as follows. The atomic propositions are ALB (Alice likes Bob), PA (Alice goes to the pub), and PB (Bob goes to the pub). We have* $\Phi_A = \{PA\}$, $\Phi_B = \{PB\}$, *and* $\Phi_E = \{ALB\}$, *with* $v_E(ALB) = \bot$, *but* $\beta_B(ALB) = \top$ *(poor deluded Bob!). For both agents* $i \in \{A, B\}$ *the cost of setting Pi to* $\top$ *is* 10, *while the cost of setting Pi to* $\bot$ *is* 0 *for both. Alice's goal is to avoid Bob:* $\gamma_A = \neg(PA \leftrightarrow PB)$. *Bob's goal is that Alice likes him, and is in the pub iff she is:* $\gamma_B = ALB \wedge (PB \wedge PA)$. *Now, it is easy to see that the game has no Nash stable state. If* $PA = PB = \bot$, *then Alice would benefit by setting* $PB = \top$, *thereby achieving her goal. If* $PA = \bot$ *and* $PB = \top$, *then Alice gets her goal achieved but Bob does not; he would do better to set* $PB = \bot$. *If* $PA = \top$ *and* $PB = \bot$, *then, again Alice gets her goal achieved but Bob does not; he would do better to set* $PB = \top$. *If* $PA = \top$ *and* $PB = \top$, *then Bob gets his goal achieved but Alice does not; she would do better to set* $PA = \bot$.

3 Announcements

Let us now return to the motivation from the introduction of the paper: namely, that a principal makes announcements about the value of environment variables in order to modify the behaviour of agents within the system. We will consider two types of announcements.

In a *simple announcement*, the principal announces the value of some non-empty subset of Φ_E to *all* the agents within the system. We will assume throughout the paper that this announcement is truthful, in that the principal does not lie about the values of the variables it announces. The effect of the announcement is that all agents within the game modify their beliefs to reflect correctly the value of the announced variables as defined in v_E. Note that we are not assuming that the principal must announce *all* variables in Φ_E: the principal is at liberty to pick some subset of Φ_E to announce. The principal will thus choose from $2^{|\Phi_E|} - 1$ possible announcements.

A *complex* announcement is more nuanced. In a complex announcement, the principal may reveal the values of different variables to different agents within the system. So, for example, the principal may reveal the value of variable p to agent i and the value of q to agent j. Again, however, we assume that announcements are truthful.

Simple Announcements: Formally, we model a simple announcement as a subset $\alpha \subseteq \Phi_E$ $(\alpha \neq \emptyset)$, with the intended meaning that, if the principal makes this announcement, then the value of every variable $p \in \alpha$ becomes common knowledge within the game. The effect of an announcement α on an agent's belief function $\beta_i : \Phi_E \to \mathbb{B}$ is to transform it to a new belief function $\beta_i \oplus \alpha$, defined as follows:

$$\beta_i \oplus \alpha(p) = \begin{cases} v_E(p) & \text{if } p \in \alpha \\ \beta_i(p) & \text{otherwise.} \end{cases}$$

With a slight abuse of notation, where G is a game and α is a possible announcement in G, we will write $G \oplus \alpha$ to denote the game obtained from G by replacing every belief function β_i in G with the belief function $\beta_i \oplus \alpha$. Observe that, given a game G and announcement α from G, computing $G \oplus \alpha$ can be done in polynomial time.

Complex Announcements: We model complex announcements as functions $\alpha : Ag \to 2^{\Phi_E}$, with the intended interpretation that after making an announcement α, an agent i comes to know the value of the environment variables $\alpha(i)$. Again, we assume truthfulness, and of course it may be that $\alpha(i) \neq \alpha(j)$. As with simple announcements, the effect of a complex announcement α on an agent is to transform its belief function β_i to a new function $\beta_i \oplus \alpha$, which in this case is defined as follows:

$$\beta_i \oplus \alpha(p) = \begin{cases} v_E(p) & \text{if } p \in \alpha(i) \\ \beta_i(p) & \text{otherwise.} \end{cases}$$

The *size* of a complex announcement α, is denoted (with a small abuse of notation) by $|\alpha|$ and is defined as: $|\alpha| = \sum_{i \in Ag} |\alpha(i)|$.

4 Announcements that Stabilise Games

Now that we have a model of announcements and their semantics, let us see how these can be used to stabilise a game. Given a game G and a simple announcement α over G, we will say that α is *stabilising* if $\mathcal{N}(G \oplus \alpha) \neq \emptyset$ (we do not require that $\mathcal{N}(G) = \emptyset$). Let $\mathcal{S}(G)$ be the set of stabilising announcements for G, i.e., $\mathcal{S}(G) = \{\alpha \mid \mathcal{N}(G \oplus \alpha) \neq \emptyset\}$.

Example 2 *We return to Alice and Bob, as discussed earlier. Suppose Alice's friend, the principal, announces $\{ALB\}$; that is, she tells Bob that Alice does not in fact like Bob. Bob updates his beliefs accordingly. At this point, Bob no longer has any possibility to achieve his desire $ALB \wedge (PB \leftrightarrow PA)$, and his optimal choice is to minimise costs by not going to the pub. Given that Bob stays at home, Alices's optimal choice to go to the pub. The outcome where $PA = \top$, $PB = \bot$ is Nash stable. Thus, $\{ALB\}$ is a stabilising announcement.*

From the point of view of the principal, the obvious decision problem relating to stabilisation is as follows: *Given a game G, does there exist some announcement α over G such that α stabilises G?* We have the following:

Proposition 1 *The problem of checking whether a game G can be stabilised by a simple announcement, (i.e., whether $\mathcal{S}(G) \neq \emptyset$), is Σ_2^p-complete; this holds even if all costs are 0.*

Proof: Membership is by the following algorithm: Guess an $\alpha \subseteq \Phi_E$ and an outcome $(v_1, \ldots, v_n)$, and verify that $(v_1, \ldots, v_n)$ is a Nash stable outcome of $G \oplus \alpha$. Guessing can clearly be done in non-deterministic polynomial time, and verification is a co-NP computation. For hardness, we reduce the problem of checking whether a Boolean game as defined in [3] has a Nash equilibrium; this was proved Σ_2^p-complete in [3]. Given a conventional Boolean game, we map the agents, goals, and controlled variables to our setting directly; we then create one new Boolean variable, call it z, and set $\Phi_E = \{z\}$. Set $v_E(z) = \top$ and $\beta_i(z) = \top$ for all agents i. Now, we claim that the system can be stabilised iff the original game has a Nash equilibrium; the only announcement that can be made is $\alpha = \{z\}$, which does not change the system in any way; the Nash stable states of the game $G \oplus \alpha$ will thus be exactly the Nash equilibria of the original game. ∎

Another obvious question is what properties announcements have. While this is not the primary subject of the present paper, it is nevertheless worth considering. We have the following:

Proposition 2 *Stability is not monotonic through announcements. That is, there exist games G and announcements α_1, α_2 over G such that $G \oplus \alpha_1$ is stable but $(G \oplus \alpha_1) \oplus \alpha_2$ is not.*

Proof: Consider the following example (a variant of the Alice and Bob example introduced earlier). Let G be the game with $Ag = \{1, 2\}$, $\Phi = \{p, q, r, s\}$, $\Phi_1 = \{p\}$, $\Phi_2 = \{q\}$, $\Phi_E = \{r, s\}$, $\beta_1(r) = \top$, $\beta_1(s) = \bot$, $\beta_2(r) = \bot$, $\beta_2(s) = \top$, $v_E(r) = \bot$, $v_E(s) = \top$, $\gamma_1 = (r \vee s) \wedge (p \leftrightarrow q)$, $\gamma_2 = \neg(p \leftrightarrow q)$, $c(p, \top) = c(q, \top) = 1$, and $c(p, \bot) = c(q, \bot) = 0$. Now, G is unstable, by a similar argument to Example 1. Announcing r will stabilise the system, again by a similar argument to Example 1. However, it is easy to see that $(G \oplus \{r\}) \oplus \{s\}$ is unstable: intuitively, in $(G \oplus \{r\})$, agent 1 does not believe he can get his goal achieved, because he believes both r and s are false, so he prefers to minimise costs by setting $p = \bot$, leaving agent 2 free to get their goal achieved by setting $q = \top$. However, in $(G \oplus \{r\}) \oplus \{s\}$, because agent 1 believes $s = \top$, he now believes he has some possibility to get his goal achieved, and the system is unstable. ∎

Let us say an announcement $\alpha \subseteq \Phi$ is *relevant* for an agent i if the announcement refers to variables that occur in the goal of i, that is, if $\alpha \cap vars(\gamma_i) \neq \emptyset$. Say α is *irrelevant* if it is not relevant for any agent. We have:

Proposition 3 *If α is irrelevant w.r.t. G then $\mathcal{N}(G) = \mathcal{N}(G \oplus \alpha)$.*

Now, an obvious question arises: Can we give some complete characterisation of games G such that $\mathcal{S}(G) \neq \emptyset$? To help understand the answer to this question, consider the following example.

Example 3 *Let $Ag = \{1, 2, 3\}$, $\Phi = \{p_1, \ldots, p_6\}$, $\Phi_i = \{p_i\}$, $\gamma_1 = p_1 \vee p_2 \vee p_4$, $\gamma_2 = p_2 \vee p_3 \vee p_5$, $\gamma_3 = p_3 \vee p_1 \vee p_6$, costs for making a variable $\top$ are 1, and for making it $\bot$ are 0. Finally, we have $\beta_i(p_j) = \bot$ for all $i \in \{1, 2, 3\}$ and $4 \leq j \leq 6$. The system is unstable: for example, the outcome in which all variables take the value $\bot$ is unstable because agent 1 could benefit by setting $p_1 = \top$. Observe, however, that any of the following announcements would serve to stabilise the system: $\alpha_1 = \{p_4\}$, $\alpha_2 = \{p_5\}$; $\alpha_3 = \{p_6\}$. For example, if announcement α_1 is made, then agent 1 will believe its goal will be achieved, and so only needs to minimise costs – it need not be concerned with what agent 2 does with p_2, which it* *does by setting $p_1 = \bot$. In this case, agent 3's best response $p_3 = \top$ (thereby achieving his goal), and agent 2 can set $p_2 = \bot$, minimising costs. This outcome is stable. Identical arguments show that α_2 or α_3 would also stabilise the system.*

How is the system stabilised in this example? Consider the announcement α_1. Before this announcement, agent 1 believes that his optimal choice may depend on the choice of agent 2; for if agent 2 were to set $p_2 = \top$ then agent 1 would prefer to set $p_1 = \bot$. In this sense, we can think of agent 1's optimal choice being *dependent* on the choice of agent 2. However, the announcement *breaks* this dependency: agent 1 now believes his goal is satisfied, and no longer needs to be concerned about the choices of others. Because his goal is satisfied, he can simply make a choice that minimises his costs: his decision becomes a (computationally very simple) optimisation problem. So, we can stabilise systems by breaking dependencies, in the way we just described. Previously, [4] studied in detail dependencies between players in Boolean games. In our context, a *dependency graph* for a game G is a digraph $D_G = (V, E)$, with vertex set $V = Ag$ and edge set $E \subseteq Ag \times Ag$ defined as follows: $(i, j) \in E$ iff $\exists(v_1, \ldots, v_j, \ldots, v_n) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_j \times \cdots \times \mathcal{V}_n$ and $v'_j \in \mathcal{V}_j$ such that $u_i(v_1, \ldots, v_j, \ldots, v_n) \neq u_i(v_1, \ldots, v'_j, \ldots, v_n)$. In other words, $(i, j) \in E$ if there is some circumstance under which a choice made by agent j can affect the utility obtained by agent i. Proposition 6 of [4] gives a sufficient condition for the existence of a Nash stable outcome, namely, if the irreflexive portion of D_G is acyclic then $\mathcal{N}(G) \neq \emptyset$. This suggests the following approach to stabilising a game G: find an announcement α such that the irreflexive portion of $D_{G\oplus\alpha}$ is acyclic. A general difficulty with this approach is implied by the following:

Proposition 4 *Given a game G and players i, j in G, the problem determining whether $(i, j) \in D_G$ is* NP*-complete.*

Proof: Membership is by "guess-and-check". For hardness, we reduce SAT. Let φ be a SAT instance. Create two agents, 1 and 2, let $\gamma_1 = \varphi \wedge z$, where z is a new variable, and let $\gamma_2 = \top$. Let $\Phi_1 = vars(\varphi)$ and $\Phi_2 = \{z\}$. All costs are 0. We now ask whether 1 is dependent on 2; we claim the answer is "yes" iff φ is satisfiable. ($\rightarrow$) Observe that the only way player 1 could obtain different utilities from two outcomes varying only in the value of z (the variable under the control of 2) is if $\varphi \wedge z$ was true in one outcome and false in the other. The outcome satisfying $\varphi \wedge z$ is then witness to the satisfiability of φ. ($\leftarrow$) If agent 1 gets the same utility for all choices it makes and choices of value for z then $\varphi \wedge z$ is not satisfiable, hence φ is not satisfiable. ∎

So, it would be helpful to identify cases where checking dependencies is computationally easy. Let us say that a goal formula γ is in *simple conjunctive form* if it is of the form $\ell_1 \wedge \cdots \wedge \ell_k$, where each ℓ_i is a literal, i.e., an atomic proposition or the negation of an atomic proposition. We assume w.l.o.g. that such a formula contains no contradictions (a proposition and its negation), as such goals are unsatisfiable. Say a game G is in simple conjunctive form if each goal γ_i is in simple conjunctive form. Then:

Proposition 5 *Suppose G is a game containing agents i and j, such that γ_i is in simple conjunctive form and $vars(\gamma_i) \cap \Phi_j \neq \emptyset$; then $(i, j) \in D_G$. It follows that, if a game is in simple conjunctive form then computing D_G can be done in polynomial time.*

So, for games in simple conjunctive form, we can easily identify the dependencies between agents. The next question is how to break these dependencies. The basic idea is, as in Example 3, to modify an agent's beliefs so that it no longer believes its optimal choice is dependent on the choices of others. We do this by convincing the agent that its goal is either guaranteed to be achieved (in which case its optimal choice is to minimise costs), or else cannot be achieved (in which case, again, the optimal choice is again simply to minimise costs). The difficulty with this approach is that we need to be careful, when making such an announcement, not to change the beliefs of other agents so that the dependency graph contains a new cycle; complex announcements will enable us to manipulate the beliefs of individual agents without affecting those of others.

Where γ_i is a goal for some agent in a game G and α is an announcement, let $\tau(\gamma_i, \alpha)$ denote the formula obtained from γ_i by systematically replacing each variable $p \in \Phi_E$ by $\beta_i \oplus \alpha(p)$. We will say α *settles* a goal γ_i if $\tau(\gamma_i, \alpha) \equiv \top$ or $\tau(\gamma_i, \alpha) \equiv \bot$. Intuitively, α settles γ_i if the result of making the announcement α is that i believes its goal is guaranteed to be true or is guaranteed to be false. With this definition in place, the following Proposition summarises our approach to stabilising games:

Proposition 6 *Suppose G is a game with cyclic dependency graph $D_G = (V, E)$, containing an edge (i, j) such that $E \setminus \{(i, j)\}$ is acyclic, and such that γ_i can be settled by some (complex) announcement α. Then G can be stabilised.*

For simple conjunctive games, we can check the conditions of Proposition 6 in polynomial time. For games in general, of course, checking the conditions will be harder.

5 Measures of Optimality for Announcements

Apart from asking whether some stabilising announcement exists, it seems obvious to consider the problem of finding an "optimal" stabilising announcement. There are many possible notions of optimality that we might consider, but here, we define just three.

Minimal Stabilising Announcements: The most obvious notion of optimality we might consider for announcements is that of *minimising size*. That is, we want an announcement α^* satisfying:

$$\alpha^* \in \arg\min_{\alpha \in \mathcal{S}(G)} |\alpha|.$$

Proposition 7 *The problem of computing the size of the smallest stabilising simple (resp. complex) announcement is in* $\text{FP}^{\Sigma_2^p[\log_2 |\Phi|]}$ *(resp.* $\text{FP}^{\Sigma_2^p[\log_2 |\Phi \times Ag|]}$*).*

Proof: We give the proof for simple announcements; the case for complex announcements is similar. Observe that the following problem, which we refer to as P, is Σ_2^p-complete using similar arguments to Proposition 1: *Given a game G, announcement α for G and $n \in \mathbb{N}$ ($n \leq |\Phi_E|$), does there exist* *a simple stabilising announcement α' for G, where $\alpha \subseteq \alpha'$, such that $|\alpha'| \leq n$?* It then follows that, for simple announcements, determining the size of the smallest stabilising announcement can be computed with $\log_2 |\Phi|$ queries to an oracle for *P* using binary search (cf. [15, pp.415–418]). ∎

Proposition 8 *The problem of computing a smallest stabilising simple (resp. complex) announcement is in* $\mathrm{FP}^{\Sigma_2^p[|\Phi|]}$ *(resp.* $\mathrm{FP}^{\Sigma_2^p[|Ag \times \Phi|]}$*).*

Proof: Compute the size *s* of the smallest announcement using the procedure of Proposition 7. Then we build a stabilising announcement α^* by dynamic programming: A variable *S* will hold the "current" announcement, with $S = \emptyset$ initially. Iteratively consider each variable $p \in \Phi_E$ in turn, invoking the oracle for *P* to ask whether there exists a stabilising announcement for *G* or size s using the partial announcement $S \cup \{p\}$; if the answer is yes, then we set $S = S \cup \{p\}$. We then move on to the next variable in Φ_E. We terminate when $|S| = s$. In this case, *S* will be a stabilising announcement of size *s*, i.e., it will be a smallest stabilising announcement. The overall number of queries to the Σ_2^p oracle for *P* is $|\Phi| + \log_2 |\Phi|$, i.e., $O(|\Phi|)$. ∎

Goal Maximising Announcements: We do not have transferable utility in our setting, and so it makes no sense to directly introduce a measure of social welfare (normally defined for an outcome as the sum of utilities of players in that outcome). However, a reasonable proxy for social welfare in our setting is to count the number of goals that are achieved in the "worst" Nash stable outcome. Formally, we want an announcement α^* satisfying:

$$\alpha^* \in \arg\max_{\alpha \in \mathcal{S}(G)} \min \\ \{succ(v_1, \ldots, v_n, v_E) \mid (v_1, \ldots, v_n, v_E) \in \mathcal{N}(G \oplus \alpha)\}.$$

Objective Satisfying Announcements: A third and final possibility, considered in [7], is the idea of modifying a game so that a particular objective is achieved in equilibrium, where the objective is represented as a formula $\Upsilon \in \mathcal{L}$. Formally, given a game G and an objective $\Upsilon \in \mathcal{L}$, we seek an announcement $\alpha^* \in \mathcal{S}(G)$ such that:

$$\forall (v_1, \ldots, v_n) \in \mathcal{N}(G \oplus \alpha^*) : (v_1, \ldots, v_n, v_E) \models \Upsilon.$$

6 Related Work & Conclusions

In the sense that the main thrust of our work is to design announcements that will modify games in such a way that certain outcomes are achieved in equilibrium, our work is similar in spirit to mechanism design/implementation theory, where the goal is to design games in such a way that certain outcomes are achieved in equilibrium [12]. However, we are aware of no work within the AI/computer science community that addresses the problem of manipulating games in the same way that we do – through communication. Work that has considered manipulating games within the AI/computer science community has focussed on the design of taxation schemes to influence behaviour [13; 2; 7]. Our work is also about the effect of making announcements, and in this sense it has some affinity with the growing body of work on *dynamic epistemic logic* (DEL) [16]. DEL tries to give a logical account of how the knowledge states of agents in a system are affected by announcements that take the form of logical formulae. Of particular interest in DEL are announcements that themselves refer to the knowledge of participants, which can affect systems in subtle and complex ways. There are *many* obvious avenues for future research. We might consider richer models of belief (possible worlds models, probabilistic and Bayesian models. . . [10]), and of course, mixed strategy equilibria. We might consider the possibility of the principal lying, and of noisy communication. We might consider the announcement of logical formulae, rather than just announcing the value of individual propositions, and we might also consider announcements that refer to the epistemic state of agents ("player one knows the value of *x*"); this would take us close to the realm of dynamic epistemic logic [16]. Finally, links with belief revision are worth examining [1].

Acknowledgments: Special thanks to the anonymous reviewer who pointed us at [4], the results of which helped us to simplify our presentation considerably. This research was financially supported by the Royal Society, MOST (#3-6797), and ISF (#1357/07).

References

[1] C. E. Alchourron, P. Gärdenfors, and D. Makinson. On the logic of theory change. *J. of Symb. Logic*, 50:510–530, 1985.

[2] Y. Bachrach, *et al.* The cost of stability in coalitional games. In *SAGT 2009*.

[3] E. Bonzon, M.-C. Lagasquie, J. Lang, and B. Zanuttini. Boolean games revisited. In *ECAI-2006*.

[4] E. Bonzon, M.-C. Lagasquie-Schiex, and J. Lang. Dependencies between players in Boolean games. *IJAR*, 50:899–914, 2009.

[5] R. Brafman, C. Domshlak, Y. Engel, and M. Tennenholtz. Planning games. In *IJCAI-2009*.

[6] P. E. Dunne, S. Kraus, W. van der Hoek, and M. Wooldridge. Cooperative boolean games. In *AAMAS-2008*.

[7] U. Endriss, S. Kraus, J. Lang, and M. Wooldridge. Designing incentives for boolean games. In *AAMAS-2011*.

[8] G. Gottlob, G. Greco, and F. Scarcello. Pure Nash equilibria: Hard and easy games. *JAIR*, 24:357–406, 2005.

[9] J. Grant, S. Kraus, and M. Wooldridge. Intentions in equilibrium. In *AAAI-2010*.

[10] J. Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2005.

[11] P. Harrenstein, W. van der Hoek, J.-J.Ch. Meyer, and C. Witteveen. Boolean games. In *TARK VIII*, 2001.

[12] E. Maskin. The theory of implementation in Nash equilibrium: A survey. MIT Department of Economics Working Paper, 1983.

[13] D. Monderer and M. Tennenholtz. K-implementation. *JAIR*, 21:37–62, 2004.

[14] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press 1994.

[15] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[16] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer-Verlag, 2007.