

Delegating Decisions in Strategic Settings

Sarit Kraus¹ and Michael Wooldridge²

Abstract. We formalise and investigate the following problem. A principal must delegate a number of decisions to a collection of agents. Once the decisions are delegated, the agents to whom the decisions are delegated will act selfishly, rationally, and independently in pursuit of their own preferences. The principal himself is assumed to be self-interested, and has some goal that he desires to be achieved. The *delegation problem* is then, given such a setting, is it possible for the principal to delegate decisions in such a way that, if all the agents to whom decisions have been delegated then make decisions rationally, the principal's goal will be achieved in equilibrium. We formalise this problem using *Boolean games*, which provides a very natural framework within which to capture the delegation problem: decisions are directly represented as Boolean variables, which the principal assigns to agents. After motivating and formally defining the delegation problem, we investigate the computational complexity of the problem, and some issues surrounding it.

1 Introduction

Throughout our lives, we must inevitably delegate to other agents decisions whose outcome will affect us, even though we know full well that the agents we delegate the decisions to are self-interested, and will make these decisions in their own interest. For example, consider the chair of a university department, who must allocate teaching and admin responsibilities to faculty members. Once the department chair has allocated duties and responsibilities, the faculty members will make the delegated decisions, but will inevitably act in their self-interest. For example, a faculty member given responsibility for designing the curriculum might favour courses in their own research area, even though these courses might not be pedagogically well motivated. A faculty member with responsibility for allocating PhD places might choose to favour his area at the expense of others. Thus, when deciding upon an allocation of decisions and responsibilities to faculty members, the faculty chair needs to take into account the choices that the faculty members will then rationally make, given their own preferences. By choosing to allocate responsibilities to different faculty members, the department chair has some leeway to influence the overall decisions that will be made. The problem the department chair then faces is how best to delegate decisions to faculty. In short, our aim in the present paper is to consider this *delegation problem* in a formal setting, and to consider the computational properties of delegation problems.

The delegation problem has some features in common with the *principal-agent problem* from economics [8, 10]. The principal-agent problem considers situations in which a principal engages the services of an agent to act on his behalf. Most consideration of the

principal-agent problem in economics has been focussed on the difficulties that arise when the actions of the agent cannot be witnessed or verified by the principal, in which case the principal cannot be certain that the agent is indeed acting in the interests of the principal. Given this asymmetry of information, a key issue is the design of a contract that will serve to align the interests of the agent with those of the principal. We will do no more here than simply note that our focus is somewhat different, and we will comment on the similarities (and differences) in the conclusions of the paper.

The formal model we use to frame the delegation problem is based on *Boolean games* [6, 2, 3, 4]. In a Boolean game, each player i is assumed to have a goal, represented as a propositional formula γ_i over some set of Boolean variables Φ . In addition, each agent i is allocated some subset Φ_i of the variables Φ , with the idea being that the variables Φ_i are under the unique control of agent i . The choices, or strategies, available to i in a Boolean game correspond to all the possible allocations of truth (\top) or falsity (\perp) to the variables Φ_i . An agent would ideally like to choose an allocation for its variables so as to satisfy its goal γ_i . Strategic concerns arise because whether i 's goal is in fact satisfied will depend on the choices made by others; and whether their goal is satisfied will in turn depend on the choice made by i , and others.

In the variant of Boolean games that we use to model the delegation problem, some of the variables in Φ may be initially unallocated, i.e., not assigned to any player's variable set Φ_i . An external *principal* (corresponding to the department chair in the example above) must allocate these variables to players within the game – that is, the decision about which unallocated variable is assigned to which player is determined by the principal. Once the principal has made an allocation, then the resulting Boolean game is played in the normal way. Thus, a player i is able to determine values for all the variables Φ_i that it is initially allocated, as well as values for the variables that were allocated to it by the principal. Note that the principal is not part of the resulting game: the values chosen for variables Φ are made by the players in the game. Thus the *only* way the principal can influence a game is in choosing the allocation of originally unallocated variables to players. The principal will make an allocation with some overall objective in mind. We represent the objective by a Boolean formula Υ over the variables Φ . If the principal is successful in allocating variables to players, then the result is that players will rationally choose values for variables so that the objective Υ is satisfied in equilibrium. Thus the overall problem faced by the principal is as follows: *Can I assign the unallocated variables to players in such a way that if the players then play the resulting game rationally, my objective Υ will be satisfied in equilibrium?* We refer to this as the *delegation problem*.

In the remainder of this paper, we formalise and study the delegation problem, focussing particularly on computational issues. After considering the basic problem, we investigate variants of it in which

¹ Department of Computer Science, Bar Ilan University, Israel. Email: sarit@cs.biu.ac.il.

² Department of Computer Science, University of Oxford, UK. Email: mjw@cs.ox.ac.uk

the principal seeks an allocation that will result in an equilibrium that maximises some objective function.

2 The Model

We now introduce the variation of Boolean games that we use in the present paper, which is directly descended from previous Boolean game models [6, 2, 3, 4].

Propositional Logic: Let $\mathbb{B} = \{\top, \perp\}$ be the set of Boolean truth values, with “ \top ” being truth and “ \perp ” being falsity. We abuse notation by using \top and \perp to denote both the syntactic constants for truth and falsity respectively, as well as their semantic counterparts. Let $\Phi = \{p, q, \dots\}$ 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 “ \perp ”. Where $\varphi \in \mathcal{L}$, we let $\text{vars}(\varphi)$ denote the (possibly empty) set of Boolean variables occurring in φ (e.g., $\text{vars}(p \wedge q) = \{p, q\}$). 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 (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 $\varphi \equiv \psi$.

Quantified Boolean Formulae: As well as propositional logic, we make use of *Quantified Boolean Formulae* (QBFs). QBFs extend propositional logic with quantifiers $\exists X$ and $\forall X$, where $X \subseteq \Phi$. A formula $\exists X : \varphi$ asserts that there is some assignment of truth values to the variables X such that φ is true under this assignment, while a formula $\forall X : \varphi$ asserts that φ is true under all assignments of truth values to the variables X . QBFs are very powerful: for example, the satisfiability of a propositional formula φ over variables Φ can be expressed as the QBF $\exists \Phi : \varphi$.

Agents and Variables: The games we consider are populated by a set $Ag = \{1, \dots, n\}$ 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 \perp) of each variable $p \in \Phi_i$. We will require that $\Phi_i \cap \Phi_j = \emptyset$ for $i \neq j$, so that no variable is controlled by more than one player. Readers who are familiar with Boolean games might also be expecting to see the requirement that every variable is controlled by an agent, but for the moment, we do not require this.

Boolean Games and Partial Games: A *partial game*, is a $(3n + 2)$ -tuple:

$$\langle Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n \rangle$$

where $Ag = \{1, \dots, n\}$ is a set of agents, $\Phi = \{p, q, \dots\}$ is a finite set of Boolean variables, $\Phi_i \subseteq \Phi$ is the set of Boolean variables under the unique control of $i \in Ag$, and $\gamma_i \in \mathcal{L}$ is the goal of agent $i \in Ag$. We will denote partial games by P, P', P_1, \dots , and let $\mathcal{P}(Ag, \Phi)$ denote the set of all partial games with agent set Ag and variable set Φ . Notice that in a partial game, it is possible that some variables in Φ are not allocated to agents, i.e., that $(\Phi_1 \cup \dots \cup \Phi_n) \subset \Phi$. Given a partial game $\langle Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n \rangle$, we let

$$\Phi_U = \Phi \setminus (\Phi_1 \cup \dots \cup \Phi_n)$$

be the set of unallocated variables. Two extremal points are worth identifying:

- First, suppose $\Phi_U = \Phi$. In this case, $\Phi_1 = \dots = \Phi_n = \emptyset$, and so *all* variables are unallocated.
- Next, suppose $\Phi_U = \emptyset$. In this case, there are *no* unallocated variables: every variable is assigned to an agent.

Where a partial game $\langle Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n \rangle$ is such that $\Phi_U = \emptyset$ then we say that $\langle Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n \rangle$ is a *Boolean game*. We use G, G_1, G', \dots to denote such Boolean games.

Choices and Outcome: 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 . 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 . The intuitive interpretation we give to \mathcal{V}_i is that it defines the *actions* or *strategies* available to agent i . An *outcome* is a collection of choices, one for each agent. Formally, an outcome for a Boolean game is a tuple $(v_1, \dots, v_n) \in \mathcal{V}_1 \times \dots \times \mathcal{V}_n$, and such an outcome uniquely defines an overall valuation for the variables Φ . We often treat outcomes for Boolean games as valuations, for example writing $(v_1, \dots, v_n) \models \varphi$ to mean that the valuation defined by the outcome (v_1, \dots, v_n) satisfies formula $\varphi \in \mathcal{L}$. Where we do not need to identify the individual components of an outcome, we will use the vector notation \vec{v} to denote outcomes, i.e., $\vec{v} = (v_1, \dots, v_n)$. Let $\text{succ}(\vec{v})$ denote the set of agents who have their goal achieved by outcome \vec{v} , i.e., $\text{succ}(\vec{v}) = \{i \in Ag \mid \vec{v} \models \gamma_i\}$.

Utility and Preference: The preferences of agents in Boolean games are very simple: an agent i strictly prefers all those outcomes that satisfy its goal γ_i over all those that do not, but is indifferent between outcomes that satisfy its goal, and is indifferent between outcomes that do not satisfy its goal. It is convenient to define for each agent $i \in Ag$ a *utility function* over outcomes which captures these preferences. The function $u_i(\dots)$ has the signature

$$u_i : \mathcal{V}_1 \times \dots \times \mathcal{V}_n \rightarrow \{0, 1\}$$

and is defined:

$$u_i(\vec{v}) = \begin{cases} 1 & \text{if } \vec{v} \models \gamma_i \\ 0 & \text{otherwise.} \end{cases}$$

Given outcomes \vec{v}_1 and \vec{v}_2 , we write $\vec{v}_1 \succeq_i \vec{v}_2$ to mean that $u_i(\vec{v}_1) \geq u_i(\vec{v}_2)$, with the corresponding strict relation \succ_i defined in the obvious way.

Nash Equilibrium: The well-known notion of Nash equilibrium [10] is readily defined for Boolean games. We say an outcome $(v_1, \dots, v_i, \dots, v_n)$ is *individually stable for agent i* if $\nexists v'_i \in \mathcal{V}_i$ such that $(v_1, \dots, v'_i, \dots, v_n) \succ_i (v_1, \dots, v_i, \dots, v_n)$. We then say an outcome (v_1, \dots, v_n) is a *Nash equilibrium* if (v_1, \dots, v_n) is individually stable $\forall i \in Ag$. We denote the Nash equilibrium outcomes of a Boolean game G by $\mathcal{N}(G)$; of course, it could be that $\mathcal{N}(G) = \emptyset$ for a given game G . We can define a Quantified Boolean Formula (QBF) that will be true in a valuation if that valuation represents a Nash equilibrium of a particular Boolean game:

$$NE(Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n) \equiv \bigwedge_{i \in Ag} ((\exists \Phi_i : \gamma_i) \rightarrow \gamma_i).$$

The key property of this definition is the following:

Proposition 1 (Bonzon et al [2]) Let G be a Boolean game and let \vec{v} be an outcome for G . Then $\vec{v} \models NE(G)$ iff $\vec{v} \in \mathcal{N}(G)$.

3 The Delegation Problem

We now come to the main problem we consider in this paper. We start with a partial game $P = \langle Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n \rangle$, with associated unallocated variable set Φ_U . The set Φ_U will represent the decisions that are to be delegated to players in the game. The allocation of Φ_U to players is done by a *principal*, who has complete freedom to allocate the variables Φ_U to players Ag as he chooses. Once the allocation is made, the partial game becomes a Boolean game, and the players will then make rational choices, resulting in some outcome. Now, the principal will in fact make the allocation with a particular *objective* in mind: we represent such an objective by a Boolean formula Υ . The idea is that the principal will try to choose an allocation so that, if the players then play the resulting Boolean game rationally, they will choose an outcome satisfying Υ .

Formally, given a partial game $P = \langle Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n \rangle$, an *allocation* is a total function

$$\alpha : \Phi_U \rightarrow Ag$$

with the intended interpretation that if $x \in \Phi_U$, then under allocation α , variable x is allocated to player $\alpha(x)$. With a small abuse of notation, we let α_i denote the set of variables allocated to player i under allocation α , i.e.,

$$\alpha_i = \{x \in \Phi_U \mid \alpha(x) = i\}.$$

Let $\mathcal{A}(P)$ denote the set of all allocations over partial game P . Where a partial game P is such that $\Phi_U = \emptyset$, we will assume the principle has a single “empty” allocation possible, and so for all partial games P , we have $\mathcal{A}(P) \neq \emptyset$. A partial game $P = \langle Ag, \Phi, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n \rangle$ together with an allocation $\alpha \in \mathcal{A}(P)$ defines a Boolean game $G(P, \alpha)$, as follows:

$$G(P, \alpha) = \langle Ag, \Phi, (\Phi_1 \cup \alpha_1), \dots, (\Phi_n \cup \alpha_n), \gamma_1, \dots, \gamma_n \rangle.$$

We let $\mathcal{G}(P)$ denote the set of Boolean games that may be obtained from partial game P through some allocation:

$$\mathcal{G}(P) = \{G(P, \alpha) \mid \alpha \in \mathcal{A}(P)\}.$$

Finally, let $\mathcal{S}(P)$ denote the set of allocations over partial game P such that under these allocations, the resulting game has a Nash equilibrium:

$$\mathcal{S}(P) = \{\alpha \in \mathcal{A}(P) \mid \mathcal{N}(G(P, \alpha)) \neq \emptyset\}.$$

Following [4], we will study two variations of the delegation problem, which we refer to as *weak* and *strong*. In the weak variation, the principal’s objective Υ is required to be satisfied in *some* Nash equilibrium of the resulting Boolean game, while in the strong variation, Υ is required to be satisfied in *all* Nash equilibria.

3.1 Weak Delegation

Formally, the WEAK DELEGATION problem is defined as follows:

Given a partial game P and an objective $\Upsilon \in \mathcal{L}$, does there exist an allocation $\alpha \in \mathcal{A}(P)$ such that Υ is satisfied in at least one Nash equilibrium of the Boolean game $G(P, \alpha)$?

We say this problem is “weak” because we only require that Υ is satisfied in *one* Nash equilibrium of $G(P, \alpha)$. We will consider stronger versions below. Notice that WEAK DELEGATION is equivalent to checking the following condition:

$$\exists G \in \mathcal{G}(P) : \exists \vec{v} \in \mathcal{N}(G) : \vec{v} \models \Upsilon.$$

The outermost existential quantifier emphasises that the task of the principal can be understood as *choosing a game* from the space of possible games $\mathcal{G}(P)$.

If $\langle P, \Upsilon \rangle$ is a positive instance of the WEAK DELEGATION problem, then we say that Υ *can be weakly implemented* in P . Following [4], we refer to weakly implementing a tautology (i.e., implementing \top where $\top \equiv \top$) as *stabilisation*. The rationale for this terminology is that weakly implementing a tautology will result in a game that has at least one Nash equilibrium – at least one steady state, in other words. It is easy to see that \top can be weakly implemented in P iff $\mathcal{S}(P) \neq \emptyset$.

We illustrate weak delegation with the following example, which we will refer to later in the paper:

Example 1 Suppose we have a partial game P with $Ag = \{1, 2\}$, $\Phi = \Phi_U = \{p, q\}$, $\Phi_1 = \Phi_2 = \emptyset$, and

$$\begin{aligned} \gamma_1 &= (p \leftrightarrow q); \\ \gamma_2 &= \neg(p \leftrightarrow q). \end{aligned}$$

Now, there are four possible allocations in $\mathcal{A}(P)$, which we will refer to as α^a through to α^d , defined as follows:

$$\begin{aligned} \alpha_1^a &= \{p, q\}; & \alpha_2^a &= \emptyset; \\ \alpha_1^b &= \{p\}; & \alpha_2^b &= \{q\}; \\ \alpha_1^c &= \{q\}; & \alpha_2^c &= \{p\}; \\ \alpha_1^d &= \emptyset; & \alpha_2^d &= \{p, q\}; \end{aligned}$$

Thus allocations α^a and α^d allocate all the variables to one player, while allocations α^b and α^c share the variables between the two players. Now, let us consider the following objectives:

$$\begin{aligned} \Upsilon_1 &= \top; \\ \Upsilon_2 &= p \vee q; \text{ and} \\ \Upsilon_3 &= p \wedge q. \end{aligned}$$

Table 1(i) illustrates which allocations of the allocations α^a – α^d weakly implement objectives Υ_1 – Υ_3 . Consider α^a first. If this allocation is made, then the resulting game has two Nash equilibria: one satisfying $p \wedge q$, the other satisfying $\neg p \wedge \neg q$. If either of allocations α^b or α^c are made, then the resulting game will have no Nash equilibria. If allocation α^d is made, then the resulting game will have two Nash equilibria: one satisfying $p \wedge \neg q$, the other satisfying $q \wedge \neg p$.

Since we have a domain with Boolean formulae, and there are clearly exponentially many allocations of variables to agents, it comes as no surprise that the WEAK DELEGATION problem is computationally hard; however, the good news is that it is no harder than the problem of determining the existence of pure strategy Nash equilibria in Boolean games:

Proposition 2 WEAK DELEGATION is Σ_2^P -complete.

Proof: Since allocations α are clearly small with respect to the size of the partial game, and since verifying that outcomes are Nash equilibria in Boolean games is in co-NP, it follows that the WEAK DELEGATION problem is in Σ_2^P : guess an allocation α and an outcome \vec{v} , and verify that both $\vec{v} \models \Upsilon$ and $\vec{v} \in \mathcal{N}(G(P, \alpha))$.

Allocation	(i) Weak Delegation			(ii) Strong Delegation			(iii) $\hat{f}_i(P, \alpha^j)$	
	Υ_1	Υ_2	Υ_3	Υ_1	Υ_2	Υ_3	f_1	f_2
α^a	y	y	n	y	n	n	0	0
α^b	n	n	n	n	n	n	N/A	N/A
α^c	n	n	n	n	n	n	N/A	N/A
α^d	y	y	y	y	y	n	1	1

Table 1. Allocations and their properties for the running example.

For hardness, we reduce the problem of checking whether a Boolean game has any pure strategy Nash equilibria, which is known to be Σ_2^P -complete [2]. Given a Boolean game G , which we wish to check for the existence pure strategy Nash equilibria, simply define the corresponding partial game P to be game G (so that $\Phi_U = \emptyset$) with objective $\Upsilon = \top$. Notice that the only possible allocation is the empty allocation, which defines the identity under the function $G(\dots)$. Now consider the WEAK DELEGATION problem:

$$\exists \alpha \in \mathcal{A}(P) : \exists \vec{v} \in \mathcal{N}(G(P, \alpha)) : \vec{v} \models \Upsilon.$$

Since $\Upsilon = \top$ and the only allocation possible is the empty allocation, this reduces to:

$$\exists \vec{v} \in \mathcal{N}(G(P, \alpha)) : \vec{v} \models \top.$$

Since the empty allocation is the identity under $G(\dots)$, and $\vec{v} \models \top$ for all \vec{v} , this further reduces to $\exists \vec{v} \in \mathcal{N}(G)$, which is exactly the problem of checking for the existence of pure strategy Nash equilibria in Boolean games. ■

Using the QBF predicate $NE(\dots)$, defined earlier, we can define a QBF formula that characterises weak delegation. Where P is a partial game and $\Upsilon \in \mathcal{L}$ is an objective, we define a QBF predicate $WI(P, \Upsilon)$ as follows:

$$WI(P, \Upsilon) \equiv \bigvee_{\alpha \in \mathcal{A}(P)} \exists \Phi : (\Upsilon \wedge NE(G(P, \alpha))).$$

We have:

Proposition 3 *Let P be a partial game and let $\Upsilon \in \mathcal{L}$ be an objective. Then the QBF formula $WI(P, \Upsilon)$ is true iff Υ can be weakly implemented in P .*

3.2 Strong Delegation

The strong version of the delegation problem requires that the objective Υ is satisfied in *all* Nash equilibria of the Boolean game that results from an allocation. Formally, the STRONG DELEGATION is defined as follows:

Given a partial game P and an objective $\Upsilon \in \mathcal{L}$, does there exist an allocation $\alpha \in \mathcal{A}(P)$ such that:

1. The Boolean game $G(P, \alpha)$ will have at least one Nash equilibrium.
2. All Nash equilibria of $G(P, \alpha)$ satisfy Υ .

Formally, this can be understood as checking the following condition:

$$\begin{aligned} \exists \alpha \in \mathcal{A}(P) : \\ \mathcal{N}(G(P, \alpha)) \neq \emptyset \ \& \\ \forall \vec{v} \in \mathcal{N}(G(P, \alpha)) : \vec{v} \models \Upsilon. \end{aligned}$$

Example 2 *Let us return to the partial game defined in Example 1. With respect to the three objectives Υ_1 – Υ_3 , Table 1(ii) summarises which can be strongly implemented. If we want to stabilise the system, then we cannot share the propositions p and q among the players: we must allocate them all to 1 or all to 2 (i.e., α^a or α^d). If we want to strongly implement Υ_2 then we must allocate all propositions to player 2. Finally, objective Υ_3 cannot be strongly implemented.*

As with weak delegation, we can characterise strong delegation in a QBF, as follows:

$$\begin{aligned} SI(P, \Upsilon) \equiv \\ \bigvee_{\alpha \in \mathcal{A}(P)} (\exists \Phi : NE(G(P, \alpha))) \wedge \\ (\forall \Phi : NE(G(P, \alpha)) \rightarrow \Upsilon) \end{aligned}$$

We have:

Proposition 4 *Let P be a partial game and let $\Upsilon \in \mathcal{L}$ be an objective. Then the QBF formula $SI(P, \Upsilon)$ is true iff Υ can be strongly implemented in P .*

4 Delegation as Optimisation

So far, we have assumed that the principal is motivated to choose an allocation so that an objective Υ is satisfied in equilibrium; the idea being that the objective represents what the principal wants to achieve through delegation. We now generalise this approach, by assuming that in delegating decisions, the principal is attempting to maximise some objective function of the form:

$$f : \mathcal{V} \rightarrow \mathbb{R}_+.$$

Thus, such an objective function assigns a positive real number $f(v)$ to every valuation $v \in \mathcal{V}$. Recalling that an outcome \vec{v} for a Boolean game corresponds to a valuation in \mathcal{V} , we will also write $f(\vec{v})$ to mean the value through f of the valuation corresponding to \vec{v} . Notice that our original formulation of delegation with respect to objective formulae Υ is a special case of the setting we are now considering, where the function f_Υ is defined for an objective formula $\Upsilon \in \mathcal{L}$ as follows:

$$f_\Upsilon(\vec{v}) = \begin{cases} 1 & \text{if } \vec{v} \models \Upsilon \\ 0 & \text{otherwise.} \end{cases}$$

The objective function f gives the value to the principal of every outcome \vec{v} . But how can we use f to obtain the value of an allocation? An allocation α for a partial game P will define a Boolean game $G(P, \alpha)$, and this game will in turn have an associated set of Nash equilibria $\mathcal{N}(G(P, \alpha))$. Our basic idea is to define the value of an allocation α for a partial game P through an objective function f to be the value of the *worst* Nash equilibrium in $\mathcal{N}(G(P, \alpha))$ [7]. However, there is a catch: what happens if $\mathcal{N}(G(P, \alpha)) = \emptyset$? In this case, we say the value of α through f is undefined. Formally, given a partial game P , allocation $\alpha \in \mathcal{A}(P)$, and objective function $f : \mathcal{V} \rightarrow \mathbb{R}_+$, we denote the value of α through f as $\hat{f}(P, \alpha)$:

1. If $\mathcal{N}(G(P, \alpha)) = \emptyset$ then $\hat{f}(P, \alpha)$ is undefined.
2. If $\mathcal{N}(G(P, \alpha)) \neq \emptyset$ then

$$\hat{f}(P, \alpha) = \min\{f(\vec{v}) \mid \vec{v} \in \mathcal{N}(G(P, \alpha))\}.$$

Now, given a partial game P and an objective function f as above, the *optimal allocation* will intuitively be the one that maximises the value of \hat{f} . Here, however, we must deal with the situation where there is no allocation that leads to a game with Nash equilibria. We will denote the optimal allocation with respect to a partial game P and objective function f by $\alpha^*(P, f)$:

1. If $S(P) = \emptyset$ then $\alpha^*(P, f)$ is undefined.
2. Otherwise $\alpha^*(P, f)$ satisfies:

$$\alpha^*(P, f) \in \arg \max_{\alpha \in S(P)} \hat{f}(P, \alpha).$$

Example 3 We work with the partial game introduced in Example 1. Consider the objective functions f_1 and f_2 , defined as follows:

$$\begin{aligned} f_1(\vec{v}) &= |\{x \in \Phi \mid \vec{v} \models x\}| \\ f_2(\vec{v}) &= |\Phi| - f_1(\vec{v}). \end{aligned}$$

Thus f_1 counts the number of variables assigned the value \top in a valuation, while f_2 counts the number of variables assigned the value \perp . Table 1(iii) summarises the properties of allocations $\alpha^a - \alpha^d$ with respect to objective functions f_1 and f_2 . Allocation α^d is thus optimal with respect to both f_1 and f_2 .

Next, we want to consider the problem of computing $\alpha^*(P, f)$. A key difficulty here is with respect to the issue of representing the objective function f . Representing the function f in a problem instance by explicitly listing all input/output pairs $(\vec{v}, f(\vec{v}))$ will not be practicable, as there will be $2^{|\Phi|}$ such pairs in total. We need a *compact* representation for f , and for the purposes of this paper, we use a well-known scheme based on weighted Boolean formulae (see, e.g., [13]).

Formally, we will say a *feature* is a pair (φ, x) , where $\varphi \in \mathcal{L}$ is a propositional formula, and $x \in \mathbb{R}_+$ is a positive real number. A *feature set*, \mathcal{F} , is simply a finite set of features, i.e., $\mathcal{F} = \{(\varphi_1, x_1), \dots, (\varphi_k, x_k)\}$. Every feature set \mathcal{F} induces an objective function $f_{\mathcal{F}}$, as follows:

$$f_{\mathcal{F}}(\vec{v}) = \sum_{\substack{(\varphi_i, x_i) \in \mathcal{F}: \\ \vec{v} \models \varphi_i}} x_i.$$

The feature set corresponding to our original objective formula Υ would be a singleton set $\{(\Upsilon, 1)\}$. Now, standard arguments from Boolean function theory tell us that: (i) the feature set representation is *complete*, in the sense that for every objective function f there exists a feature set \mathcal{F} such that $f = f_{\mathcal{F}}$; (ii) the feature set representation is *more compact* than the explicit representation for many objective functions f ; however (iii) there are objective functions for which the smallest equivalent feature set will be broadly of the same size (i.e., will require exponentially many features).

We have the following results relating to the computation of optimal allocations.

Proposition 5 Given a partial game P , feature set \mathcal{F} , and allocation α , the problem of determining whether $\hat{f}_{\mathcal{F}}(P, \alpha)$ is defined is Σ_2^P -complete.

Proof: This is directly equivalent to determining whether the game $G(P, \alpha)$ has a Nash equilibrium. ■

Proposition 6 Given a partial game P and feature set \mathcal{F} , the problem of determining whether $\alpha^*(P, f_{\mathcal{F}})$ is defined is Σ_2^P -complete.

Proof: This is directly equivalent to checking whether $S(P) \neq \emptyset$, which is Σ_2^P -complete by the proof of Proposition 2. ■

Now, the OPTIMAL DELEGATION problem is the following function problem:

Given a partial game P and feature set \mathcal{F} , output:

0 if $\alpha^*(P, f_{\mathcal{F}})$ is undefined.

1 · $\alpha^*(P, f_{\mathcal{F}})$ for some optimal allocation $\alpha^*(P, f_{\mathcal{F}})$ if such an allocation is defined.

In the decision variant of the problem, we are given a target value $k \in \mathbb{R}_+$ and asked whether, if $\alpha^*(P, f_{\mathcal{F}})$ is defined, we have $\hat{f}_{\mathcal{F}}(P, \alpha^*(P, f_{\mathcal{F}})) \geq k$. From the arguments above, this problem is Σ_2^P -complete. We then have:

Proposition 7 Given a partial game P and feature set \mathcal{F} , the OPTIMAL DELEGATION problem is in $FP^{\Sigma_2^P}$.

Proof: Check whether $\alpha^*(P, f_{\mathcal{F}})$ is defined, and if not return 0. Otherwise, define a value μ as follows:

$$\mu = \sum_{(\varphi_i, x_i) \in \mathcal{F}} x_i$$

That is, μ is the largest value that an optimal allocation could possibly take. We thus have:

$$0 \leq \hat{f}_{\mathcal{F}}(P, \alpha^*(P, f_{\mathcal{F}})) \leq \mu.$$

We can then use binary search to find the value $\hat{f}_{\mathcal{F}}(P, \alpha^*(P, f_{\mathcal{F}}))$ of an optimal allocation $\alpha^*(P, f_{\mathcal{F}})$, by invoking a Σ_2^P oracle for the decision variant of the problem. We start by asking whether $\hat{f}_{\mathcal{F}}(P, \alpha^*(P, f_{\mathcal{F}})) \geq \mu/2$; if the answer is no, then we ask whether $\hat{f}_{\mathcal{F}}(P, \alpha^*(P, f_{\mathcal{F}})) \geq \mu/4$, while if the answer is yes, we ask whether $\hat{f}_{\mathcal{F}}(P, \alpha^*(P, f_{\mathcal{F}})) \geq (3\mu)/4$, and so on. We will converge to the value of the optimal allocation with at most polynomially many queries to the decision variant of the problem (cf. [12, p.416]). Given the value of the optimal allocation, we can then find an optimal allocation with at most a further $|Ag \times \Phi|$ queries to a Σ_2^P oracle. ■

5 Related Work

The delegation problem is closely related to the *principal-agent problem* studied in economics [8, 10]. A typical setting for the principal agent problem is where a principal engages the services of an agent to work on behalf of the principal, typically for a fee. The basic issue studied in the principal-agent problem is that the agent will be self-interested, and it may not be feasible for the principal to observe the actions of the agent; in which case, how can the principal be certain that the agent is indeed acting in the principals interests? Typical solutions to the principal-agent problem involve designing incentive schemes that will help to align the preferences of the agent with those of the principal. In commercial companies, this is exactly the role of profit-sharing schemes such as stock options. Although it is closely related, the issues studied in the principal-agent problem are somewhat distinct from those considered in the present paper.

Our work is also relevant to *logics of propositional control* [15, 5, 14]. Originally developed by van der Hoek and Wooldridge [15], and extended by Gerbrandy [5] these logics are specifically intended

to reason about scenarios in which a collection of agents each have control over some set of Boolean variables. Logics of propositional control extend classical propositional logic with indexed modal operators $\diamond_C \varphi$, where C is a set of agents; the intended interpretation of this expression is that the agents in C can assign values to their variables in such a way as to satisfy φ . Of specific relevance to the present paper is the logic DCL-PC [14], which makes it possible to reason about situations where control of Boolean variables is transferred between agents. For example, the DCL-PC formula

$$[1 \rightsquigarrow_p 2] \diamond_{2p} p \wedge q$$

asserts that “after agent 1 gives control of variable p to 2, then agent 2 can choose values for its variables so that $p \wedge q$ is true”. Here, $1 \rightsquigarrow_p 2$ is the action of transferring control of p from agent 1 to agent 2. Actions such as $1 \rightsquigarrow_p 2$ in DCL-PC can be used to directly model the delegation of decisions in the setting we have considered in this paper. More precisely, given a partial game P and an objective Υ , it is possible to define a DCL-PC model \mathcal{M}_P and DCL-PC formulae $\varphi^\exists(P, \Upsilon)$ and $\varphi^\forall(P, \Upsilon)$ such that (i) $\mathcal{M}_P \models \varphi^\exists(P, \Upsilon)$ iff Υ can be weakly implemented in P , while (ii) $\mathcal{M}_P \models \varphi^\forall(P, \Upsilon)$ iff Υ can be strongly implemented in P . In this way, we could use the logic DCL-PC as a formal system for reasoning about delegation problems. The exact details are somewhat technical, however, and well beyond the scope of the present paper.

We should also point to other formalisms for reasoning about delegation [9, 11, 1]. However, in these other works, the focus is rather different to our own. For example, Li *et al.* focus on decentralized trust management [9], while Norman and Reed consider a logic of delegation based on the STIT (“see to it that”) operator [11].

Finally, we remark that it would be interesting to consider mapping our delegation problems into standard extensive form games of incomplete information [10, pp.221–228]. Given a partial game P and objective Υ , the corresponding extensive form game contains the principal as a player, as well as the player set Ag from P . Since Boolean games contain simultaneous moves, (the players simultaneously choose values for their variables) we need to eliminate these in the extensive form game representation. We can do this by assuming an ordering of the players ρ , which will represent the order in which players make their moves. Given P, Υ, ρ , the extensive form game $\Gamma_{P, \Upsilon, \rho}$ is defined as follows:

1. The principal moves first, choosing an assignment $\alpha \in \mathcal{A}(P)$.
2. For each value $i \in \mathbb{N}$ from 1 to $|Ag|$, player $\rho[i]$ then makes a choice for the variable set $\Phi_i \cup \alpha_i$ (if player i has no variables for which to allocate values, then i makes a “dummy” move, with an empty allocation).
3. The information set for player $\rho[i]$, $1 < i \leq |Ag|$, is such that i knows the move of the principal, but does not know the choices made by players $\rho[j]$, $1 \leq j < i$, (although $\rho[i]$ will know who precedes it in the ordering ρ , and hence which players have already made choices).
4. Player $\rho[1]$ and the principal have perfect information.
5. Terminal nodes of the game tree correspond to choices \vec{v} , and payoffs are assigned in terminal nodes so that the principal gets payoff 1 in a terminal node corresponding to \vec{v} if $\vec{v} \models \Upsilon$, and other players $i \in Ag$ get payoff 1 if $\vec{v} \models \gamma_i$ and payoff 0 otherwise.

The appropriate solution concept for analysing such games is Perfect Bayesian Equilibrium [10, p.285]. We leave the analysis for future work. Note that, while it could be arguably useful to consider delegation problems as extensive form games, (since this provides us with

a battery of standard game theoretic techniques that we can bring to bear on the analysis of such problems), we feel that the formulation presented in the present paper represents a more natural model of the problem at hand: we directly represent decisions to be delegated as Boolean variables, and directly model who they are being allocated to.

6 Conclusions & Future Work

We have introduced and investigated the problem of how to optimally delegate decisions to self-interested agents. We modelled this delegation problem using the setting of Boolean games. We argue that Boolean games provide a very natural framework within which to model the delegation of Boolean decisions: individual decisions naturally map to Boolean variables, owned by individual agents. In future work, as we noted in the preceding section, it would be interesting to consider in more detail the delegation problem as an extensive form game. The use of logical systems such as QBF and DCL-PC to analyse delegation would also be worth pursuing in more detail; and of course, more fine grained complexity analysis would be welcome.

Acknowledgments: Kraus was supported by the Google Inter-university center for Electronic Markets and Auctions and MOST (#3-6797). Wooldridge was supported by ERC Advanced Grant 291528 (RACE).

REFERENCES

- [1] G. Boella and L. van der Torre, ‘Delegation of power in normative multi-agent systems’, in *Proc. DEON 2006*.
- [2] E. Bonzon, M.-C. Lagasque, J. Lang, and B. Zanuttini, ‘Boolean games revisited’, in *Proc. ECAI-2006*.
- [3] P. E. Dunne, S. Kraus, W. van der Hoek, and M. Wooldridge, ‘Cooperative boolean games’, in *Proc. AAMAS-2008*.
- [4] U. Endriss, S. Kraus, J. Lang, and M. Wooldridge, ‘Designing incentives for boolean games’, in *Proc. AAMAS-2011*.
- [5] J. Gerbrandy, ‘Logics of propositional control’, in *Proc. AAMAS-2006*.
- [6] P. Harrenstein, W. van der Hoek, J.-J.Ch. Meyer, and C. Witteveen, ‘Boolean games’, in *Proc. TARK VIII*, (2001).
- [7] E. Koutsoupias and C. Papadimitriou, ‘Worst-case equilibria’, in *Sixteenth Annual Symposium on Theoretical Aspects of Computer Science (STACS-99)*, pp. 404–413, (1999).
- [8] J.-J. Laffont and D. Martimort, *The Theory of Incentives: The Principal-Agent Model*, Princeton University Press, 2001.
- [9] N. Li, B. N. Grosz, and J. Feigenbaum, ‘Delegation logic: A logic-based approach to distributed authorization’, *ACM Transactions on Information and System Security*, **6**(1), 128 – 171, (2003).
- [10] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*, Oxford UP, 1995.
- [11] T. J. Norman and C. Reed, ‘Group delegation and responsibility’, in *Proc. AAMAS-2002*.
- [12] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.
- [13] J. Uckelman, Y. Chevaleyre, U. Endriss, and J. Lang, ‘Representing utility functions via weighted goals’, *Math. Logic Quarterly*, **55**(4), 341–361, (2009).
- [14] W. van der Hoek, D. Walther, and M. Wooldridge, ‘Reasoning about the transfer of control’, *JAIR*, **37**, 437–477, (2010).
- [15] W. van der Hoek and M. Wooldridge, ‘On the logic of cooperation and propositional control’, *Artificial Intelligence*, **64**, 81–119, (2005).