Incentives in Effort Games

Yoram Bachrach Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel
{yori, jeff}@cs.huji.ac.il

Abstract

We consider *Effort Games*, a game theoretic model of cooperation in open environments, which is a variant of the principal-agent problem from economic theory. In our multiagent domain, a common project depends on various tasks; achieving certain subsets of the tasks completes the project successfully, while others do not. The probability of achieving a task is higher when the agent in charge of it exerts effort, at a certain cost for that agent. A central authority, called the principal, attempts to incentivize agents to exert effort, but can only reward agents based on the success of the entire project.

We model this domain as a normal form game, where the payoffs for each strategy profile are defined based on the different probabilities of achieving each task and on the boolean function that defines which task subsets complete the project and which do not. We view this boolean function as a simple coalitional game, and call this game the underlying coalitional game. We consider the computational complexity of testing whether exerting effort is a dominant strategy for an agent and of finding a reward strategy for this domain, using either a *dominant strategy equilibrium* or using *iterated elimination of dominated strategies*. We show these problems are generally #P-hard, and that they are at least as computationally hard as calculating the Banzhaf power index in the underlying coalitional game. We also show that in a certain restricted domain, where the underlying coalitional game is a weighted voting game with certain properties, it is possible to solve all of the above problems in polynomial time.

1 Introduction

The computational aspects of many game theoretic concepts have been thoroughly studied in recent years, as they are significant for many real-world domains, including auctions, voting, and electronic commerce. A key issue in many such domains is constructing a proper reward scheme to achieve the desired behavior of self-interested agents.

There are many such examples: auction theory is sometimes used to design auctions that maximize the revenue of the auctioneer; social choice theory attempts to design mechanisms that give agents incentives to truthfully reveal their preferences so that an optimal social choice is made; solution concepts in coalitional game theory are used to make sure that rewards are distributed in a fair manner, or so that the coalitions that are formed are stable. The computational aspects of such concepts are critical, since in order to use them in practice, one must find a tractable way to compute them.

1.1 Our Setting

In this paper, we deal with the computational complexity of finding a reward scheme in *effort games* in open environments. In our model, the mechanism's purpose is to incentivize agents to exert effort when working on a common project. Completing the project requires the performance of various

tasks. Some of the subsets of the tasks are "winning" subsets, so that when these tasks are completed the project succeeds, and some subsets are "losing", so when only these tasks are completed, the project fails.

A self-interested agent is in charge of each task. This agent can exert effort, which increases the probability that its task will be completed successfully. However, this exertion of effort has a certain cost for the agent. On the other hand, the task has a certain probability of being completed successfully even if the agent in charge of it does not exert any effort. A natural way to reward agents is based on the effort they have exerted. However, in open environments this information is sometimes not available to the mechanism.

One example of such a situation is maintenance efforts. Consider, for example, a communication network with a source node and a target node, where agents are in charge of maintenance tasks for the links between the nodes of the network. If no maintenance effort is expended on a link, it has a some probability $\alpha < 1$ of functioning, but if a maintenance effort is made for the link, it functions with a higher probability $\beta > \alpha$. Consider a mechanism in charge of sending some information between source and target. This mechanism only knows whether it succeeded in sending the information, but if that attempt fails, may not know which links failed or if they failed due to lack of maintenance. Another example is voting domains, which we consider in Section 4.

In the above examples, the interested party, called *the principal*, cannot observe the agents' decisions about whether to expend effort. The only information available to it is the overall result of the project. However, the agents have a certain cost to exerting efforts, and require a certain reward as an incentive for making such an effort. The principal cannot offer any reward based on whether an agent has exerted effort or not, since it does not have that information. It can, however, offer the agents a certain reward only if the project is successful. Since agents have a certain capability of increasing the probability of obtaining this outcome, they may exert effort in order to get their reward. However, they would only do so if the expected reward from exerting effort is greater than the cost of that effort for them.

The principal thus offers a reward vector $r = (r_1, \ldots, r_n)$, where for all i, $r_i \geq 0$. If the project succeeds, the reward of agent a_i is r_i, and if the project fails then for all a_i the reward of a_i is 0. Consider a principal that wants to motivate a certain subset of the agents to exert effort. Of course, it can offer each agent in this subset a reward that is high enough to make sure it is worthwhile for them to exert effort no matter what the other agents do. On the other hand, the principal wants to minimize the total rewards given when the project succeeds, $\sum_{i=0}^{n} r_i$.

In this paper, we examine the computational complexity of finding out whether a certain reward vector makes it the dominant strategy of a certain agent to exert effort, and the computational complexity of constructing a reward vector that guarantees a certain subset of agents exert effort. The paper proceeds as follows. In Section 2 we present several required game theoretic notions, and give the formal model of effort games. In Section 3 we present the main results regarding the computational complexity of calculating a successful reward scheme, and in Section 4 we present algorithms for restricted types of weighted voting games. In Section 5 we discuss related work and similar problems. We conclude in Section 6.

2 The Formal Model

2.1 Preliminaries

We now define several game theoretic notions required for defining effort games. The definition of effort games relies on the definition of both normal form games and simple coalitional games.

Definition 1. *An n-player normal form game is given by a set of agents (players) $I = \{a_1, ..., a_n\}$, and for each agent a_i a (finite) set of pure strategies S_i, and a utility (payoff) function $F_i : S_1 \times S_2 \times ... \times S_n \rightarrow \mathbb{R}$. $F_i(s_1, ..., s_n)$ denotes a_i's utility when each player a_j plays strategy s_j.*

For brevity, we denote the set of strategy profiles $\Sigma = S_1 \times S_2 \times ... \times S_n$, and denote items in Σ as $\sigma \in \Sigma$ ($\sigma = (s_1, ..., s_n)$, where $s_i \in S_i$). We

also denote $\Sigma_{-i} = S_1 \times ... \times S_{i-1} \times S_{i+1} \times ... \times S_n$, and given an incomplete strategy profile $\sigma' = (s_1, ..., s_{i-1}, s_{i+1}, ..., s_n) \in \Sigma_{-i}$ we denote $(\sigma_{-i}, s_i) = (s_1, ..., s_{i-1}, s_i, s_{i+1}, ...s_n) \in \Sigma$.

Given a normal form game G, we say agent a_i's strategy $s_x \in S_i$ strongly dominates $s_y \in S_i$ if a_i's utility is higher when using s_x than when using s_y, no matter what strategies the other agents use.

Definition 2. *Strictly dominating strategy. Given a normal form game G, we say agent a_i's strategy $s_x \in S_i$ strictly dominates $s_y \in S_i$ if the following holds: For any incomplete strategy profile $\sigma_{-i} \in \Sigma_{-i}$ we have $F_i((\sigma_{-i}, s_x)) > F_i((\sigma_{-i}, s_y))$.*

Definition 3. *Dominant Strategy. Given a normal form game G, we say agent a_i's strategy s_x is a_i's dominant strategy if it dominates all other strategies $s_i \in S_i$.*

Given a certain game, game theoretic solution concepts specify which outcomes are reasonable, under various assumptions of rationality and common knowledge. In this paper, we will deal with two solution concepts: dominant strategy equilibrium, and iterated elimination of dominated strategies. Given a certain game, an agent a_i may or may not have a dominant strategy. One known game theoretic solution concept is a dominant strategy equilibrium. This concept defines a *single* reasonable strategy profile (and derived payoffs) for certain games.

Definition 4. *Dominant Strategy Equilibrium. Given a normal form game G, we say a strategy profile $\sigma = (s_1, ..., s_n) \in \Sigma$ is a dominant strategy equilibrium if for any agent a_i, strategy s_i is a dominant strategy for a_i.*

Another solution concept is that of *iterated elimination of dominated strategies*. In iterated dominance, strictly dominated strategies are removed from the game, and no longer have any effect on future dominance relations. A certain strategy $s_x \in S_i$ may not dominate $s_y \in S_i$, because s_y performs better against $\sigma_a \in \Sigma_{-i}$. However, if one of the strategies in σ_a is removed, that profile is no longer applicable, so s_x may now dominate s_y. It is a well-known fact that if we remove dominated strategies until no more dominated strategies remain, in the end the remaining strategies for each player will be the same, regardless of the order in which strategies are removed [10]. Some authors refer to this as the fact that iterated (strict) dominance is path-independent.

The heart of an effort game is a boolean function that decides which task subsets complete the common project successfully, and which task subsets do not. Our results depend on viewing this boolean function as a simple coalitional game. We now define a simple coalitional game.[1]

Definition 5. *A coalitional game is a domain that consists of a set of tasks, T, and a characteristic function mapping any subset of the tasks to a real value $v : 2^T \to \mathbb{R}$, indicating the total utility of a project that achieves exactly these tasks.*

In a *simple* coalitional game, v only gets values of 0 or 1 ($v : 2^I \to \{0, 1\}$). We say a subset $C \subset T$ *wins* if $v(T) = 1$, and say it *loses* if $v(C) = 0$. We denote the set of all subsets of tasks that win the simple game as $T_{win} = \{T' \subset T | v(T') = 1\}$. A task t is *critical* in a winning subset C if the task's removal from that coalition makes it lose: $v(C) = 1$, $v(C \setminus \{t\}) = 0$. A game is *increasing* if for all subsets $C' \subset C \subset T$ we have $v(C') \leq v(C)$. We will assume achieving more tasks is always better for the project, so games are increasing. Thus, if a certain subset of tasks $C \subset T$ wins, every superset of C also wins.

We now define the weighted voting game, which is a famous game-theoretic model of cooperation in political bodies. In this game, each agent has a weight, and a coalition of agents wins the game if the sum of the weights of its members exceeds a certain threshold.

Definition 6. *A weighted voting game is a simple coalitional game with tasks (agents) $T = (t_1, ..., t_n)$, a vector of weights $w = (w_1, ..., w_n)$ and a threshold q. We say t_i has the weight w_i. Given a coalition $C \subseteq T$ we denote the weight of the coalition $w(C) = \sum_{i \in \{i | t_i \in C\}} w_i$. A coalition C wins the game*

[1] Rather than defining the function over agents, as typically done in definitions for coalitional games, we call the players in this game tasks, to keep our terminology consistent with the rest of the paper.

(so $v(C) = 1$) if $w(C) \geq q$, and loses the game (so $v(C) = 0$) if $w(C) < q$.

A question that arises in the context of simple games, and especially weighted voting games, is that of measuring the influence a certain task (player) has on the outcome of the game. One approach to measuring this notion is power indices. A possible definition of the power of an agent is its *a priori* probability of having a significant impact on the outcome of the game. Different definitions of "significant impact" have resulted in the definition of different power indices, one of which is the Banzhaf index [3].

Definition 7. *The Banzhaf index is a power index that depends on the number of coalitions in which an agent is critical, out of all possible coalitions. It is given by $\beta(v) = (\beta_1(v), ..., \beta_n(v))$ where*

$$\beta_i(v) = \frac{1}{2^{n-1}} \sum_{S \subset T | i \in S} [v(S) - v(S \setminus \{i\})].$$

The Banzhaf power index reflects the assumption that the agents are independent in their choices. Another prominent power index, the Shapley-Shubik power index does not reflect such an assumption. This property of the Banzhaf index is similar to some of the assumptions of effort games.

2.2 The Effort Game Model

We now define effort game models and related problems.

Definition 8. *An effort game domain is a domain that consists of the following: A set of n agents, $I = \{a_1, ..., a_n\}$; a set of n tasks, $T = \{t_1, ..., t_n\}$; a simple coalitional game G with task set T, such that $|T| = n$, and with the value function $v : 2^T \rightarrow \{0, 1\}$; a set of success probability pairs $(\alpha_1, \beta_1), (\alpha_2, \beta_2), ..., (\alpha_n, \beta_n)$ so that $\alpha_i \in \mathbb{R}$, and that $0 \leq \alpha_i \leq \beta_i \leq 1$; a set of effort exertion costs $c_1, ..., c_n \in \mathbb{R}$, so that $c_i > 0$.*

Informally, this domain is interpreted as follows. A joint project depends on the completion of certain tasks. Achieving some of the subsets of tasks completes the project successfully, and some fail, as determined by the simple game G. An agent a_i is responsible for each such task t_i. That agent may exert effort, which gives the task a probability of β_i to be completed. However, exerting effort costs that agent a certain utility c_i. If the agents shirks (does not exert effort) the agent does not incur the cost c_i, and the task has a lower probability α_i of being completed.

We now formally describe the domain. Each of the agents $a_i \in I$ is responsible for the task $t_i \in T$. For each coalition of agents $C \subset I$, we denote the set of tasks owned by these agents $T(C) = \{t_i \in T | a_i \in C\}$. We say the common project is successful if a subset of tasks $T' \subset T$ is achieved, so that $v(T') = 1$ (so T' is winning in G). Each agent can either choose to exert effort or shirk. In the effort games model, we assume the tasks succeed or fail *independently* of one another. If a_i exerts effort, task t_i is completed with probability β_i. If it does not exert effort (and shirks instead), task t_i is completed with a lower probability $\alpha_i < \beta_i$. However, each agent has a cost for exerting effort, $c_i > 0$, which is deducted from the utility obtained by the agent. Suppose the agents in C exert effort, and that the agents in $I \setminus C$ do not. Given C we know the probability that each task is completed: if $a_i \in C$ then t_i is completed with probability β_i. If $a_i \notin C$, t_i is completed with probability α_i.

Given the coalition of agents that contribute effort, C, we denote the probability that a certain task t_i is completed as $p_i(C)$, defined as follows:

$$p_i(C) = \begin{cases} \beta_i & \text{if } t_i \in C \\ \alpha_i & \text{if } t_i \notin C \end{cases}$$

Consider a subset of tasks $T' \subset T$. Given the coalition C of agents that exert effort, we can calculate the probability that exactly the tasks in T' are the ones achieved: $\Pr_C(T') = \prod_{t_i \in T'} p_i(C) \cdot \prod_{t_i \notin T'} (1 - p_i(C))$. We can calculate the probability that *any* winning subsets of tasks is achieved, and denote this by $Pr_C(Win) = \sum_{T_w \in T_{win}} Pr_C(T_w)$.

In our model we have a central authority (called the *principal*) interested in successfully completing the common project. The principal attempts to make sure a certain subset of agents exert effort, and needs to reward the agents so they will exert effort despite their cost of doing so. He thus designs a *reward*

scheme, but attempts to minimize his costs.

Let $C \subset I$ be the coalition of agents that have exerted effort, and T' be the set of achieved tasks. On the one hand, T' may not contain all the tasks of the agents that have exerted effort, since if $\beta_i < 1$, a task has a probability of failing even when the agent exerts effort. On the other hand, T' may contain some tasks for which agents did not exert effort, since if $\alpha_i > 0$, a task has a probability of succeeding even if an agent does not exert effort. We assume that the principal knows whether the project succeeded or not (whether $v(T') = 1$ or $v(T') = 0$), knows c_i, α_i, β_i of all the agents, but does not know whether an agent a_i has exerted effort (so $a_i \in C$) or shirked (so $a_i \notin C$). Thus it cannot reward only those agents that have exerted effort. It can only promise each agent a_i a certain reward r_i if the project succeeds, and a reward of 0 if it does not. The principal can choose among various reward vectors $r = (r_1, ..., r_n)$.

Given the reward vector $r = (r_1, ..., r_n)$, and given that the agents that exert effort are $C \subset I$, a_i's expected reward is $e_i(C) = \sum_{T_w \in T_{win}} \Pr_C(T') \cdot r_i$. Agent a_i has a cost c_i of exerting effort. It can choose between two strategies—exert effort, or shirk. Exerting effort increases the expected reward, but has a cost c_i. If a_i shirks he does not incur the cost c_i, but his expected reward is smaller. The effort game is the normal form game obtained due to a certain reward vector r chosen by the principal.

Definition 9. *An Effort Game is the normal form game $G_e(r)$ defined on the above domain with a simple coalitional game G and a reward vector $r = (r_1, ..., r_n)$, as follows.*

In $G_e(r)$ agent a_i has two strategies: $S_i = \{exert, shirk\}$. Denote by Σ the set of all strategy profiles $\Sigma = S_1 \times ... \times S_n$. Given a strategy profile $\sigma = (s_1, ...s_n) \in \Sigma$, we denote the coalition of agents that exert effort in σ by $C_\sigma = \{a_i \in I | s_i = exert\}$. To fully define the game, we must also define the payoff function of each agent $F_i : \Sigma \to \mathbb{R}$. The payoffs depend on the reward vector $r = (r_1, ..., r_n)$: the payoff of each agent in strategy profile σ is his expected reward minus the cost of the effort exerted. Thus:

$$F_i(\sigma) = \begin{cases} e_i(C_\sigma) - c_i & \text{if } s_i = exert \\ e_i(C_\sigma) & \text{if } s_i = shirk \end{cases}$$

G_e depends on r, so we denote it $G_e(r)$.

Given a simple coalitional game G, each reward vector r defines a different effort game $G_e(r)$. Given an effort game, the principal may want to make sure a certain subset of the agents exert effort, and it is up to him to choose a reward vector that achieves this, under certain assumptions on the rational behavior of these agents. The strategies used by the agents are determined by a certain *game theoretic solution concept*. Such solution concepts typically define different possible strategy profiles. A reward vector that guarantees that a certain coalition C' exerts effort, under a certain solution concept, is an incentive inducing scheme for C'.[2]

Definition 10. *Incentive Inducing Scheme for a Coalition C' in the Effort Game G_e is a reward vector $r = (r1, ..., r_n)$, such that in the effort game $G_e(r)$, in* any *strategy profile $\sigma \in \Sigma$ allowable by the solution concept, the agents in C' exert effort, so $C' \subset C_\sigma$.*

Although there may be many possible incentive inducing reward vectors, the principal is self-interested, and attempts to minimize the total of rewards it pays, $\sum_{i=1}^{n} r_i$.

Given an effort game domain G_e and a coalition of agents $C' \subset I$ that the principal wants to exert effort, we now define two types of incentive inducing schemes: a dominant strategy scheme, and an iterated elimination of dominated strategies scheme.

Definition 11. *A Dominant Strategy Incentive Inducing Scheme for C' is a reward vector $r = (r1, ..., r_n)$, s.t. for any $a_i \in C'$, exerting effort is a dominant strategy for a_i.*

[2]Several papers regarding the "combinatorial agency" [1] have focused on a Nash equilibrium domain. We survey some of this work in Section 5. In this paper, we focus on a dominant strategy implementation, and on an iterated elimination of dominant strategies implementation.

Definition 12. *An Iterated Elimination of Dominated Strategies Incentive Inducing Scheme for C' is a reward vector $r = (r1, ..., r_n)$, such that in the effort game $G_e(r)$, after any sequence of eliminating dominated strategies, for any $a_i \in C'$, the only remaining strategy for a_i is to exert effort.*

2.2.1 Strategy Changes and Expected Rewards

We first define the following relation regarding effort exertion in strategy profiles. Let D be an effort game domain, and $r = (r1, \ldots, r_n)$ be a reward vector. Let $\sigma_1, \sigma_2 \in \Sigma$ be two strategy profiles in $G_e(r)$, so $\sigma_1 = (s_{1,1}, s_{1,2}, \ldots, s_{1,n})$ and $\sigma_2 = (s_{2,1}, s_{2,2}, \ldots, s_{2,n})$. We say σ_1 is *more exerting* than σ_2, and denote $\sigma_1 >_e \sigma_2$ if the following holds: all the agents that exert effort in σ_2 also exert effort in σ_1, and at least one agent that exerts effort in σ_1 does not exert effort in σ_2, so for all a_i we have $s_{2,i} = \textit{exert} \Rightarrow s_{1,i} = \textit{exert}$, and for some a_j we have $s_{2,j} = \textit{shirk}$ and $s_{1,j} = \textit{exert}$.

We now show that the expected reward of a given agent increases when more of the other agents exert effort (no matter whether that agent exerts effort himself or not). We first prove that if a *single* agent that shirked decides to exert effort, the expected rewards of all agents increase.

Theorem 1. *Let D be an effort game domain, $r = (r1, \ldots, r_n)$ be a reward vector, and a_i be a certain agent in that domain. Let $\sigma_1, \sigma_2 \in \Sigma$ be two strategy profiles in $G_e(r)$ so that for all $a_j \neq a_i$ we have that $\sigma 1, j = \sigma_{2_j}$, and that $\sigma 1, i = \textit{exert}$ and $\sigma 2, i = \textit{shirk}$. Then for all j we have that $e_j(C_{\sigma_1}) > e_j(C_{\sigma_2})$.*

Proof. The only difference between σ_1 and σ_2 is the strategy used by a_i, who shirks in σ_2 but exerts effort in σ_1. Agent a_i is in charge of task t_i. C_σ is the set of agents that exert effort in strategy profile σ, so $C_{\sigma_1} = C_{\sigma_2} \cup \{a_i\}$.

Consider a task subset T_x. If $t_i \notin T_x$, denote $T_x^d = T_x \cup \{t_i\}$, and if $t_i \in T_x$ denote $T_x^d = T_x \setminus \{t_i\}$. The notation in Section 2.2 denoted the probability that exactly the tasks in T_x are the ones achieved as $\Pr_C(T_x) = \prod_{t_i \in T_x} p_i(C) \cdot \prod_{t_i \notin T_x}(1 - p_i(C))$. We note that if $t_i \in T_x$ then $\Pr_{C_{\sigma_1}}(T_x) = \prod_{t_j \in (T_x \setminus \{t_i\})} p_j(C_{\sigma_1}) \cdot \prod_{t_j \in (T \setminus T_x \setminus \{t_i\})}(1 - p_i(C_{\sigma_1})) \cdot \beta_i > \prod_{t_j \in (T_x \setminus \{t_i\})} p_j(C_{\sigma_1}) \cdot \prod_{t_j \in (T \setminus T_x \setminus \{t_i\})}(1 - p_i(C_{\sigma_1})) \cdot \alpha_i = \Pr_{C_{\sigma_2}}(T_x)$. Thus, if $t_i \in T_x$ then $\Pr_{C_{\sigma_1}}(T_x) > \Pr_{C_{\sigma_2}}(T_x)$.

We denote the probability of completing T_x given a *partial* strategy profile σ_{-i} (with a missing strategy for a_i) as $\Pr_{C_{\sigma_{-i}}}(T_x) = \prod_{t_j \in (T_x \setminus \{t_i\})} p_j(C) \cdot \prod_{t_j \in (T \setminus T_x \setminus \{t_i\})}(1 - p_j(C))$. We note that this probability ignores the question of whether t_i was completed (even if $t_i \in T_x$) and only takes into consideration the tasks in $T_x \setminus \{t_i\}$. Given this notation, we can see that for any task subset T_x we have that $\Pr_{C_{\sigma_1}}(T_x) + \Pr_{C_{\sigma_1}}(T_x^d) = \Pr_{C_{\sigma_2}}(T_x) + \Pr_{C_{\sigma_2}}(T_x^d)$, since: $\Pr_{C_{\sigma_1}}(T_x) + \Pr_{C_{\sigma_1}}(T_x^d) = \Pr_{C_{\sigma_{-i}}}(T_x) \cdot \beta_i + \Pr_{C_{\sigma_{-i}}}(T_x) \cdot (1 - \beta_i) = \Pr_{C_{\sigma_{-i}}}(T_x) \cdot \alpha_i + \Pr_{C_{\sigma_{-i}}}(T_x) \cdot (1 - \alpha_i) = \Pr_{C_{\sigma_2}}(T_x) + \Pr_{C_{\sigma_2}}(T_x^d)$.

In Section 2.2 we defined the expected reward as $e_i(C_\sigma) = \sum_{T_w \in T_{win}} \Pr_C(T_w) \cdot r_i$. We have assumed the underlying coalitional game G is increasing (Section 2), so if a task subset T_a wins, so does any superset of it, so if $T_a \subset T_b$ then $T_a \in T_{win} \Rightarrow T_b \in T_{win}$. Consider a *winning* task subset $T_w \in T_{win}$. Either $t_i \in T_w$ or $t_i \notin T_w$. If $t_i \notin T_w$, then T_w^d is also winning and $T_w^d \in T_{win}$ since $T_w \in T_{win}$ and G is increasing (and $T_w \subset T_w^d$). Thus, for any winning task subset $T_w \in T_{win}$ either $t_i \in T_w$ or T_w^d is also winning, so $T_w^d \in T_{win}$. Since $e_k(C_\sigma) = r_k \cdot \sum_{T_w \in T_{win}} \Pr_{C_\sigma}(T_w)$, we can express it as:
$e_k(C_\sigma) = r_k \cdot ((\frac{1}{2} \sum_{T_w \in \{T' \in T_{win} | T' \in T_{win} \wedge T'^d \in Twin\}} (\Pr_{C_\sigma}(T_w) + \Pr_{C_\sigma}(T_w^d))) + \sum_{T_w \in \{T' \in T_{win} | t_i \in T'\}} Pr_{C_\sigma}(T_w))$. Since when moving from C_{σ_2} to C_{σ_1} the first sum remains the same and the second increases, we have that $e_k(C_{\sigma_1}) > e_k(C_{\sigma_2})$. We chose a_k to be any agent, so this holds for all agents. □

We now show that the expected reward of any agent increases when more of the other agents exert effort.

Theorem 2. *Let D be an effort game domain, $r = (r1, \ldots, r_n)$ be a reward vector, and a_i be an agent in that domain. Let $\sigma_1, \sigma_2 \in \Sigma$ be two strategy profiles in $G_e(r)$ so that $\sigma_1 >_e \sigma_2$, and so that $\sigma_{1,i} = \sigma_{2_i}$. Then $e_i(C_{\sigma_1}) > e_i(C_{\sigma_2})$.*

Proof. Since $\sigma_1 >_e \sigma_2$, the only difference between

σ_1 and σ_2 are several agents that exert effort in σ_1 but shirk in σ_2. It is thus possible to create a series of strategy profiles $\sigma_2 <_e \sigma'_1 <_e \sigma'_2 <_e \ldots <_e \sigma'_k <_e \sigma_1$, such that σ'_i and σ'_{i+1} are identical, except for one agent that shirks in σ'_i but exerts effort in σ'_{i+1}. We can then apply Theorem 1, and get that $e_i(C_{\sigma'_{i+1}}) > e_i(C_{\sigma'_i})$, and that $e_i(C_{\sigma_1}) > e_i(C_{\sigma_2})$. □

The above Theorems 1 and 2 have considered the expected rewards given a certain strategy profile, but can also be stated with regard to the probability of having a winning task subset.

Corollary 1. *If $\sigma_1, \sigma_2 \in \Sigma$ are strategy profiles such that for all $a_j \neq a_i$ we have that $\sigma 1, j = \sigma_{2_j}$, and that $\sigma 1, i = exert$ and $\sigma 2, i = shirk$, then for all j we have that $\sum_{T_w \in T_{win}} \Pr_{C_{\sigma_1}}(T_w) > \sum_{T_w \in T_{win}} \Pr_{C_{\sigma_2}}(T_w)$. If $\sigma_1 >_e \sigma_2$ are strategy profiles such that $\sigma_1 >_e \sigma_2$ and such that $\sigma_{1,i} = \sigma_{2_i}$, then $\sum_{T_w \in T_{win}} \Pr_{C_{\sigma_1}}(T_w) > \sum_{T_w \in T_{win}} \Pr_{C_{\sigma_2}}(T_w)$.*

Proof. We simply consider Theorems 1 and 2 for the case where $r_i = 1$. □

3 The Complexity of Incentives

Given an effort game, agents naturally consider whether they should exert effort, and the principal naturally considers how it should incentivize a certain subset of the agents to exert effort, while minimizing the sum of rewards it must give. Different assumptions the principal has about the rational behavior of the agents are reflected in the solution concept it uses. We now formally frame the above problems. In the rest of this section, we consider an effort game domain D, with agents $I = \{a_1, \ldots, a_n\}$, where each a_i is responsible for task $t_i \in$. The underlying coalitional game is G, with the value function $v : 2^T \rightarrow \{0, 1\}$. The set of success probabilities is $(\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n)$, and the effort exertion costs are $c_1, \ldots, c_n$.

The following problems concern a reward vector $r = (r_1, \ldots, r_n)$, the effort game $G_e(r)$, and a target agent a_i.

Definition 13. *DSE (DOMINANT STRATEGY EXERT): Given $G_e(r)$, is "exert" a dominant strategy for a_i?*

Definition 14. *IEE (ITERATED ELIMINATION EXERT): Given $G_e(r)$, is "exert" the* only *remaining strategy for a_i after iterated elimination of dominated strategies? This means that if Σ' is the set of strategy profiles remaining after a sequence of iterated elimination, then for any strategy profile $\sigma' = (\sigma'_1, \ldots, \sigma'_n) \in \Sigma'$ we have $\sigma'_i = exert$.*

The following problems concern the effort game domain D, and a coalition C.

Definition 15. *D-INI (DOMINANT INDUCING INCENTIVES): Given D, compute a dominant strategy incentive inducing scheme $r = (r1, \ldots, r_n)$ for C (see Definition 11).*

Definition 16. *IE-INI (ITERATED ELIMINATION INDUCING INCENTIVES): Given D, compute an iterated elimination of dominated strategies incentive inducing scheme $r = (r1, \ldots, r_n)$ for C (see Definition 12).*

The computational complexity of these problems is investigated in the following sections.

3.1 Rewards and the Banzhaf Power Index

We now show the relation between the complexity of the above problems, and power indices in the underlying coalitional game.

Theorem 3. *Let D be an effort game domain, where for a_i we have $\alpha_i = 0$ and $\beta_i = 1$, and for all $a_j \neq a_i$ we have $\alpha_j = \beta_j = \frac{1}{2}$. Let $r = (r_1, \ldots, r_n)$ be a reward vector, so that for a_i exerting effort is a dominant strategy in $G_e(r)$. Then, $r_i > \frac{c_i}{\beta_i(v)}$ (where $\beta_i(v)$ is the Banzhaf power index of t_i in the underlying coalitional game G, with the value function v).*

Proof. Agent a_i only has two strategies in $G_e(r)$: $S_i = \{exert, shirk\}$. As in Definition 9, a_i's payoff in $G_e(r)$ when exerting effort is $e_i(C_\sigma) - c_i$, and $e_i(C_\sigma)$ when not exerting effort. Given an incomplete

strategy profile σ^a_{-i} (with a strategy missing for a_i), we denote its completion with a_i exerting effort as $\sigma^a_e = (\sigma_{-i}, s_i = \textit{exert})$, and its completion when a_i shirks as $\sigma^a_s = (\sigma_{-i}, s_i = \textit{shirk})$. σ^a_e and σ^a_s are two complete strategy profiles, which are identical in all strategies, except that of a_i.

For exerting effort to be a dominant strategy, a_i's payoff when exerting effort must be greater than its payoff when shirking, no matter what the other agents do. Thus, for exerting effort to be a dominant strategy the following must hold: for *all* incomplete strategy profiles σ^a_{-i} we have that:

$$e_i(C_{\sigma^a_e}) - c_i > e_i(C_{\sigma^a_s}).$$

We can restate this as:

$$\sum_{T_w \in T_{win}} r_i \Pr_{C_{\sigma^a_e}}(T_w) - c_i > \sum_{T_w \in T_{win}} r_i \Pr_{C_{\sigma^a_s}}(T_w)$$

or more briefly as:

$$r_i > \frac{c_i}{\sum_{T_w \in T_{win}} (\Pr_{C_{\sigma^a_e}}(T_w) - \Pr_{C_{\sigma^a_s}}(T_w))}$$

It remains to show that $\sum_{T_w \in T_{win}} (\Pr_{C_{\sigma^a_e}}(T_w) - \Pr_{C_{\sigma^a_s}}(T_w)) = \beta_i(v)$.

Let $T_w \in T_{win}$ be a task subset so that $t_i \in T_w$. Since $\alpha_i = 0$ and $\beta_i = 1$ and since for any $j \neq i$ we have $\alpha_j = \beta_j = \frac{1}{2}$, for any σ^a_{-i} we have $\Pr_{C_{\sigma^a_e}}(T_w) = (\frac{1}{2})^{n-1}$ and $\Pr_{C_{\sigma^a_s}}(T_w) = 0$.

On the other hand, let $T_w \in T_{win}$ be a task subset so that $t_i \notin T_w$. Due to the same reason, for any σ^a_{-i} we have $\Pr_{C_{\sigma^a_e}}(T_w) = 0$ and $\Pr_{C_{\sigma^a_s}}(T_w) = (\frac{1}{2})^{n-1}$.

Thus we have:

$\sum_{T_w \in T_{win}} (\Pr_{C_{\sigma^a_e}}(T_w) - \Pr_{C_{\sigma^a_s}}(T_w)) =$
$\sum_{T_w \in T_{win} | t_i \in T_w} (\Pr_{C_{\sigma^a_e}}(T_w) - \Pr_{C_{\sigma^a_s}}(T_w)) +$
$\sum_{T_w \in T_{win} | t_i \notin T_w} (\Pr_{C_{\sigma^a_e}}(T_w) - \Pr_{C_{\sigma^a_s}}(T_w)) =$
$\sum_{T_w \in T_{win} | t_i \in T_w} ((\frac{1}{2})^{n-1} - 0) +$
$\sum_{T_w \in T_{win} | t_i \notin T_w} (0 - (\frac{1}{2})^{n-1}) =$
$(\frac{1}{2})^{n-1} \cdot [\sum_{T_w \in T_{win} | t_i \in T_w} 1 - \sum_{T_w \notin T_{win} | t_i \in T_w} 1] = \beta_i(v)$

Note that the final equality requires that v is increasing, so every winning coalition C that $a_i \notin C$ also wins when a_i is added to that coalition, so $C \cup \{a_i\}$ also wins. □

Consider an effort game domain D, the reward vector r, and the resulting effort game $G_e(r)$. We now show that DSE, testing whether exerting effort is a dominant strategy for a certain agent a_i, is at least as hard computationally as calculating the Banzhaf power index in the underlying coalitional game G. Since calculating the Banzhaf index is known to be #P-hard in various domains (see Section 5), this shows that in general DSE is also #P-hard.

Theorem 4. *DSE is as hard computationally as calculating the Banzhaf power index of its underlying coalitional game G.*

Proof. Consider a simple coalitional game G with a characteristic function v. We reduce the problem of calculating $\beta_i(v)$, the Banzhaf index of agent t_i in G, to a polynomial number of DSE problems. There are 2^{n-1} possible values $\beta_i(v)$ can take: $\frac{1}{2^{n-1}}, \frac{2}{2^{n-1}}, \frac{3}{2^{n-1}}, \ldots, 1$. We perform a binary search on the correct value of β_i, by using DSE queries. We construct an effort game domain, that contains an agent a_j for each task t_j in G. The success probabilities for a_i are $\alpha_i = 0$ and $\beta_i = 1$. For all other agents $a_j \neq a_i$ the success probabilities are $\alpha_j = \beta_j = \frac{1}{2}$. The effort exertion costs are $c_i = 1$ for all the agents.

The first DSE query is regarding the reward vector of $r = (r_1 = 0, r_2 = 0, \ldots, r_{i-1} = 0, r_i = 2 \cdot c_i, r_{i+1} = 0, \ldots, r_n = 0)$. Due to Theorem 3, if the DSE answer is "yes", the following must hold: $r_i = 2 \cdot c_i > \frac{c_i}{\beta_i(v)}$, or equivalently $\beta_i(v) > \frac{1}{2}$. Similarly, to test if $\beta_i > \frac{1}{k}$, we simply test DSE with the reward vector $r_1 = 0, r_2 = 0, \ldots, r_{i-1} = 0, r_i = k \cdot c_i, r_{i+1} = 0, \ldots, r_n = 0$. Thus, we are able to find β_i in $O(\log_2(2^{n-1})) = O(n)$, so a polynomial procedure for DSE can be used to construct a polynomial procedure for calculating the Banzhaf power index in the underlying game. □

One domain of coalitional games where calculating the Banzhaf power index is known to be NP-hard is weighted voting games [5]. Thus, in an effort game where the underlying coalitional game is a weighted voting game, it is NP-hard to test if exerting effort is a dominant strategy. Obviously, calculating a reward vector that will make the dominant strategy of all the agents be "exert effort" is harder than testing whether a given reward vector makes exerting effort

a dominant strategy for some agent, so D-INI is also generally NP-hard.

4 Inducing Incentives in Weighted Voting Games

We now consider a restricted class of effort games in weighted voting domains, and show how to find a dominant strategy incentive inducing scheme, or an iterated elimination of dominated strategies incentive inducing scheme, in these domains. The domain highlights the relation between power in the underlying coalitional game, and the incentives for the resulting effort game domain.

In our domain, voters decide on a course of action using weighted voting. Each voter has a weight, and a decision passes if the total weight of the agents that vote for it exceeds a certain threshold. We will consider the case where the voters may have different weights (voter i has weight w_i), but the quota is so high that in fact the decision can only pass when *all* of the agents vote for it: we assume that the quota for passing the decision is $q = \sum_{i=1}^{n} w_i$. Thus, in this restricted setting, all the voters have equal power (so the Banzhaf power index is the same for all the agents, even though they have different weights).

Suppose each voter has a probability of α to vote in favor of the decision. An agent a_i may increase the probability of voter v_i voting in favor of the decision to $\beta > \alpha$, at a certain cost of exerting this effort, c_i. In our domain, we will assume the effort exertion cost is proportional to the voter's weight, so $c_i = w_i$. Consider a principal, that wants to maximize the probability of this decision passing. In this case, the principal wants all the agents to exert their effort, so all the voters would have a high probability of accepting the decision. Formally we model this situation as an effort game domain D. The underlying coalitional game G has the tasks $T = (t_1, \ldots, t_n)$. A coalition of tasks $C \subseteq T$ wins in G if it contains all the tasks and loses otherwise, so $v(T) = 1$, and for all $C \neq T$ we have $v(C) = 0$. The success probability pairs are identical for all tasks: $(\alpha_1 = \alpha, \beta_1 = \beta), (\alpha_2 = \alpha, \beta_2 = \beta), \ldots, (\alpha_n = \alpha, \beta_n = \beta)$. Agent a_i is in charge of t_i, and the effort exertion costs are the weights in the underlying weighted voting game, so $c_i = w_i$.

Consider the above effort game domain D. Given a reward vector $r = (r_1, \ldots, r_n)$ we get the effort game $G_e(r)$. We show how to compute both a dominant strategy incentive inducing scheme (D-INI) and an iterated elimination of dominated strategies incentive inducing scheme (IE-INI) in this domain. We also show how to find such an IE-INI vector that minimizes $\sum_{i=1}^{n} r_i$.

We first show how to calculate the minimal reward r_i that makes exerting effort a dominant strategy for a_i in the above domain.

Lemma 1. *If $r_i > \frac{c_i}{\alpha^{n-1} \cdot (\beta - \alpha)}$, then exerting effort is a dominant strategy for a_i.*

Proof. As seen in Theorem 3, the condition for exerting effort being a dominating strategy for a_i is that for every $\sigma^a \in \Sigma_{-i}$: $r_i > \frac{c_i}{\sum_{T_w \in T_{win}} (\Pr_{C_{\sigma^a_e}}(T_w) - \Pr_{C_{\sigma^a_s}}(T_w))}$. However, in this domain, the only winning task subset is T, so we can restate this as: $r_i > \frac{c_i}{(\Pr_{C_{\sigma^a_e}}(T) - \Pr_{C_{\sigma^a_s}}(T))}$. Since the only difference between σ^a_e and σ^a_e is that a_i exerts effort in the first and shirks in the second, we have $\Pr_{C_{\sigma^a_e}}(T) = P_{\sigma^a(T)} \cdot \beta$ and $\Pr_{C_{\sigma^a_s}}(T) = P_{\sigma^a(T)} \cdot \alpha$ (the notation $P_{\sigma^a(T)}$ denotes the probability of completing T given a *partial* strategy profile $\sigma^a \in \Sigma_{-i}$ similar to the notation used in Theorem 1). Thus, an equivalent condition for having exerting effort be the dominant strategy for a_i is that for every $\sigma^a \in \Sigma_{-i}$ we have $r_i > \frac{c_i}{\Pr_{C_{\sigma^a}}(T) \cdot (\beta - \alpha)}$. Each $\sigma^a \in \Sigma_{-i}$ thus requires a different minimal reward r_i, and the smaller $\Pr_{C_{\sigma^a}}(T)$, the higher r_i must be. In our domain, $\Pr_{C_{\sigma^a}}(T)$ is smallest when all the agents shirk in σ^a, so $C_{\sigma^a} = \phi$ and $\Pr_{C_{\sigma^a}}(T) = \alpha^{n-1}$. □

The above lemma allows us to solve D-INI in this domain in polynomial time—we have a simple formula for the minimal reward vector r which is a dominant strategy incentive inducing scheme.

Corollary 2. *D-INI is in P for the effort game in the above specific weighted voting domain. The following reward vector is a dominant strategy incentive inducing scheme: $r^* = (\frac{c_1}{\alpha^{n-1} \cdot (\beta - \alpha)}, \ldots, \frac{c_n}{\alpha^{n-1} \cdot (\beta - \alpha)})$.*

Proof. A D-INI solution is a reward vector that makes exerting effort the dominant strategy for each of the agents. Due to Lemma 1, r^* is indeed such a vector. We also note that if $r_i < \frac{c_i}{\alpha^{n-1}\cdot(\beta-\alpha)}$, shirking is the better strategy for a_i under the strategy profile where all the other agents shirk, so the above r^* is the dominant strategy incentive inducing scheme which is *minimal* in terms of $\sum_{i=1}^{n} r_i$. This direct formula can be calculated in polynomial time. □

We now consider IE-INI, computing an iterated elimination of dominated strategies incentive inducing scheme in this domain. We show that such a scheme can significantly reduce the total rewards $\sum_{i=1}^{n} r_i$. We suggest an IE-INI procedure for this domain.

Due to Lemma 1, if $r_i > \frac{c_i}{\alpha^{n-1}\cdot(\beta-\alpha)}$, then exerting effort is a dominant strategy for a_i. Thus, after one step of elimination of dominated strategies, a_i is sure to exert effort in the game. Consider an agent a_j, who knows a_i would exert effort. We denote the set of strategy profiles which miss both the strategies for a_i and a_j as $\Sigma_{-\{i,j\}} = \Sigma_1 \times \ldots \times S_{i-1} \times S_{i+1} \times \ldots S_{j-1} \times S_{j+1} \times S_n$. Similarly to the notation in Theorem 1) we denote the probability of completing $T_x \subseteq T$ given a *partial* strategy profile $\sigma^b \in \Sigma_{-\{i,j\}}$ = (missing the strategies of both a_i and a_j) as $\Pr_{C_{\sigma^b}}(T_x) = \prod_{t_k \in (T_x \setminus \{t_i,t_j\})} p_k(C) \cdot \prod_{t_k \in (T \setminus T_x \setminus \{t_i,t_j\})} (1 - p_k(C))$. We note that this probability ignores the question of whether t_i or t_j were completed (even though $t_i, t_j \in T_x$) and only takes into consideration the tasks in $T_x \setminus \{t_i, t_j\}$. The following lemma shows that the fact that one agent is sure to exert effort makes it cheaper to make sure another agent exerts effort.

Lemma 2. *Let a_i, a_j be two agents,and r_i, r_j be their rewards so that $r_i > \frac{c_i}{\alpha^{n-1}\cdot(\beta-\alpha)}$ and that $r_j > \frac{c_j}{\alpha^{n-2}\cdot\beta\cdot(\beta-\alpha)}$. Then under iterated elimination of dominated strategies, the only remaining strategy for both a_i and a_j is to exert effort.*

Proof. We follow steps similar to Lemma 1. Since a_i exerts effort, the condition for exerting effort being a dominating strategy for a_j (after eliminating shirking as a strategy for a_i) is that for every $\sigma^b \in \Sigma_{-\{i,j\}}$ we have $r_j > \frac{c_j}{\Pr_{C_{\sigma^b}(T)\cdot\beta\cdot(\beta-\alpha)}}$. Similarly to Theorem 3, each $\sigma^b \in \Sigma_{-\{i,j\}}$ requires a different minimal reward r_j, and the smaller $\Pr_{C_{\sigma^b}}(T)$, the higher r_j must be. Again, $\Pr_{C_{\sigma^b}}(T)$ is smallest when all the agents shirk in σ^b, so $C_{\sigma^b} = \phi$ and $\Pr_{C_{\sigma^b}}(T) = \alpha^{n-2}$. Thus, if a_i is known to exert effort, a reward $r_j > \frac{c_j}{\alpha^{n-2}\cdot\beta\cdot(\beta-\alpha)}$ guarantees that exerting effort is a dominant strategy for a_j (when a_i is known to exert effort). □

We now consider an iterated elimination of dominated strategies incentive inducing scheme (IE-INI). We first choose an ordering of the agents. We then go through the agents in that order, and find the minimal reward required to make exerting effort a dominant strategy for each of them, given that its predecessors exert effort as well. Denote by π a permutation (reordering) of the agents (so $\pi : \{1,\ldots,n\} \to \{1,\ldots,n\}$ and π is reversible). We denote the set of all such permutations Π. $\pi(i)$ is the location of a_i in the new ordering of the agents. In the generated reward scheme, the strategy "shirk" for $a_{\pi(i)}$ is eliminated during round i of strategy elimination.

Theorem 5. *IE-INI is in P for the effort game in the above mentioned specific weighted voting domain. For any reordering of the agents $\pi \in \Pi$, a reward vector r_π where for all agents a_i we have: $r_{\pi(i)} > \frac{c_{\pi(i)}}{\alpha^{n-\pi(i)}\cdot\beta^{\pi(i)-1}\cdot(\beta-\alpha)}$ is an iterated elimination incentive inducing scheme.*

Proof. The above definition requirement for the first agent in the reordering, a_i such that $\pi(i) = 1$, is exactly as required in Lemma 1, and for the second agent exactly as in Lemma 2. We can continue the process: given that $r_i > \frac{c_i}{\alpha^{n-1}\cdot(\beta-\alpha)}$ and that $r_j > \frac{c_j}{\alpha^{n-2}\cdot\beta\cdot(\beta-\alpha)}$, we can calculate the minimal reward r_k that makes exerting effort a dominant strategy for yet another agent, a_k. Similarly to the proof of Lemma 2, we can see that $r_k > \frac{c_k}{\alpha^{n-3}\cdot\beta^2\cdot(\beta-\alpha)}$. By induction and using the same proof technique as in Lemma 2, we can see that $r_{\pi(i)} > \frac{c_{\pi(i)}}{\alpha^{n-\pi(i)}\cdot\beta^{\pi(i)-1}\cdot(\beta-\alpha)}$ makes "exert" a dominant strategy for $a_{\pi(i)}$ after i rounds of eliminating dominated strategies. This direct formula can be calculated in polynomial time. □

We note that each reordering π of the agents results in a different reward vector. The principal is interested in minimizing $\sum_{i=1}^{n} r_i$. We show that the best

ordering is according to the agents' exertion costs $c_i = w_i$.

Lemma 3. *The reward vector r_π that minimizes $\sum_{i=1}^{n} r_i$ is achieved by sorting agents by their weights $w_i = c_i$, from smallest to biggest.*

Proof. Let $a = (a_{i_1}, a_{i_2}, \ldots, a_{i_{k-1}}, a_{i_k}, a_{i_{k+1}}, a_{i_{k+2}}, \ldots, a_{i_n})$ and $b = (a_{i_1}, a_{i_2}, \ldots, a_{i_{k-1}}, a_{i_{k+1}}, a_{i_k}, a_{i_{k+2}}, \ldots, a_{i_n})$ be two reorderings of the agents that are identical except switching places between the adjacent a_{i_k} and $a_{i_{k+1}}$. Let $r^a = (r^a_1, \ldots, r^a_n), r^b = (r^b_1, \ldots, r^b_n)$, be the resulting reward vectors for these orderings, as in Theorem 5. It suffices to show that if $w_{i_k} > w_{i_{k+1}}$ then $\sum_{i=1}^{n} r^b_i < \sum_{i=1}^{n} r^a_i$, since if this holds, we have a "bubble sort" sequence of switches of agent pairs that monotonically drops the total rewards paid, and ends in the ordering of the agents according to their size.

Consider an agent a_i which is in the k'th location in the sequence. That is, the rewards for all the agents in locations $1, 2, ..., k-1$ are such that after $k-1$ rounds of elimination of dominated strategies, they are all sure to exert effort.

Due to Theorem 5, the required reward for it is $r_i > \frac{c_i}{\alpha^{n-k} \cdot \beta^{k-1} \cdot (\beta - \alpha)}$. Moving it to the $k+1$ location would make this required reward be $r_i > \frac{c_i}{\alpha^{n-k-1} \cdot \beta^{k} \cdot (\beta - \alpha)}$. The ratio between the two required rewards is $\frac{\alpha}{\beta} < 1$.

Denote by A the required reward for a_{i_k} when he is on the k'th location, so Aq is the required reward when he is in the $k+1$'th location. Denote by B the required reward for $a_{i_{k+1}}$ when he is on the k'th location, so Bq is the required reward when he is in the $k+1$'th location. Thus switching the agent's location in the sequence changes the total rewards from $A + Bq$ to $Aq + B$. We note that $A > B$, due to the equation of Theorem 5. Since $q < 1$ and since $A > B$ the change in reward $A + Bq - Aq + B$ is positive, so we can improve the principal's situation by switching these agents. Thus, $w_{i_k} > w_{i_{k+1}}$, and then $\sum_{i=1}^{n} r^b_i < \sum_{i=1}^{n} r^a_i$. □

We have thus shown that in this domain we can find an IE-INI by sorting the agents according to their weights (and thus costs of exerting efforts), and using the equation from Theorem 5 to construct the reward vector. Section 3.1 discussed effort games over *general* weighted voting games, and showed that in that domain it is NP-hard to find an IE-INI, since it is NP-hard to compute the Banzhaf power index in weighted voting games. However, this section discusses weighted voting games where the threshold is so high that the only winning coalition is that of all the agents. In this domain, all agents have the same Banzhaf power index, which can be calculated using a direct formula, and indeed, for this domain we have shown that IE-INI is in P.

5 Related Work

The effort games model relies on both cooperative and non-cooperative game theoretic models. Dominant strategy equilibria, Nash equilibria, and iterated elimination of strictly dominated strategies are all known solution concepts. However, the computational complexity of iterated elimination of dominated strategies has been less studied than the complexity of Nash equilibria. A few papers that do deal with this issue are [6, 8, 4].

The concept of power indices originated from work on cooperative game theory. The Shapley-Shubik index [12], an application of the Shapley value [11], has been used to measure power in committees and politics. The Banzhaf power index originated in [3]. Several papers have dealt with the computational complexity of calculating power indices. [9] has shown that calculating the Banzhaf index in weighted voting games is NP-complete. [2] has considered the problem of calculating the Banzhaf power index in *network flow games*. In this coalitional game, which models a certain problem in network reliability, agents control links in a network flow graph, and a coalition wins if it manages to allow a certain flow between a source vertex and a target vertex. It was shown for that domain that calculating the Banzhaf index is #P-complete.

Our model of effort games considers the fact that agents working in teams may not exert effort when their decisions are unobservable. This creates a moral hazard problem. Such considerations are also discussed in [7].

A model very similar to ours appears in [13]. That

paper considers a simple situation, where a set of identical agents are working on a common project. Each agent may exert effort or not. Similarly to our model, each agent is in charge of a certain task, which has a probability of $\beta = 1$ of succeeding when effort is exerted, and some probability α of succeeding even if the agent shirked. However, that paper defines a very *restricted* effort game, where in the underlying coalitional game the only winning coalition is the grand coalition I, where α is the same for all agents, and where a task is always completed when the agent in charge exerts effort, so $\beta = 1$. [13] considers a principal who attempts to incentivize all the agents to exert effort, and shows a certain, easily calculable, reward vector which is an iterative elimination of dominated strategies implementation. Since agents in that domain are identical, yet the rewards are different, it shows that a certain discrimination allows the principal to minimize his payments while still having the project succeed with probability of 1. Our model of effort games extends that model, by allowing different subsets of the tasks to complete the project. The focus of [13] is on the economic question of discrimination in such settings, whereas our paper concentrates on the computational features of effort games, and their relation to power indices.

A model similar to ours is also given in [1]. That work also considers a domain where a principal employs agents in a joint project. The common theme of this paper and [1] is that the actions taken by the agents affect the probability that the project is successful. However, the model there is more general—it allows agents to take more than one action, and each strategy profile results in a certain distribution over the possible outcomes for the joint project. One major difference between our model and their model is the choice of a solution concept. [1] focuses on a Nash Equilibrium, while this paper focuses on the stronger notion of a dominant strategy equilibrium and on iterated elimination of dominated strategies equilibrium. Thus, our probablistic model can be represented within "combinatorial agency". However, we have different assumptions about the rational behavior of the agents, which are reflected in the solution concept used. These different solution concept changes the complexity of inducing incentives, so our results are very different from those in that line of related work.

6 Conclusions and Future Work

We defined the effort games model of cooperation, discussed how an effort game relies on an underlying coalitional game, and showed that the complexity of inducing incentives (or even simply testing if exerting effort is a dominant strategy) is at least as hard as calculating the Banzhaf power index in the underlying game. We also discussed effort games where the underlying coalitional game is a restricted class of weighted voting games, and showed that for this domain we can answer several questions regarding the incentives in polynomial time. Defining the relation between computing incentives in an effort game and calculating the Banzhaf index in the underlying game allows us to easily use complexity hardness results regarding power indices to show complexity results in effort games.

It remains an open problem to examine the complexity of inducing incentives for other classes of underlying games. It also remains an open problem to consider the relation between the computational complexity of inducing effort in effort games, and the computational complexity of other problems in the underlying coalitional game. For example, we believe that inducing incentives in effort games is at least as hard computationally as various counting problems. Given our hardness results for general effort games, it will also be interesting to see if incentive inducing schemes can be approximated in polynomial time.

References

[1] M. Babaioff, M. Feldman, and N. Nisan. Combinatorial agency. In *ACM EC'06*, 2006.

[2] Y. Bachrach and J. S. Rosenschein. Computing the Banzhaf power index in network flow games. In *AAMAS'07*, pages 323–329, Honolulu, May 2007.

[3] J. F. Banzhaf. Weighted voting doesn't work: a mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[4] V. Conitzer and T. Sandholm. Complexity of (iterated) dominance. In *EC-2005*, pages 88–97, 2005.

[5] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.

[6] I. Gilboa, E. Kalai, and E. Zemel. The complexity of eliminating dominated strategies. *MOR: Mathematics of Operations Research*, 18, 1993.

[7] B. Holmstrom. Moral hazard in teams. *Bell Journal of Economics*, 13(2):324–340, 1982.

[8] D. E. Knuth, C. H. Papadimitriou, and J. N. Tsitsiklis. A note on strategy elimination in bimatrix games. *Operations Research Letters*, 7(3):103–107, 1988.

[9] Y. Matsui and T. Matsui. NP-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science*, 263(1–2):305–310, 2001.

[10] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1999.

[11] L. S. Shapley. A value for n-person games. *Contrib. to the Theory of Games*, pages 31–40, 1953.

[12] L. S. Shapley and M. Shubik. A method for evaluating the distribution of power in a committee system. *American Political Science Review*, 48:787–792, 1954.

[13] E. Winter. Incentives and discrimination. *American Economic Review*, 94(3):764–773, 2004.