

Combinatorial Agency

[Extended Abstract] *

Moshe Babaioff
School of Information
Management and Systems
UC Berkeley
Berkeley, CA, 94720 USA
moshe@sims.berkeley.edu

Michal Feldman
School of Engineering and
Computer Science
The Hebrew University of
Jerusalem
Jerusalem, 91904 Israel
mfeldman@cs.huji.ac.il

Noam Nisan
School of Engineering and
Computer Science
The Hebrew University of
Jerusalem
Jerusalem, 91904 Israel
noam@cs.huji.ac.il

ABSTRACT

Much recent research concerns systems, such as the Internet, whose components are owned and operated by different parties, each with his own “selfish” goal. The field of Algorithmic Mechanism Design handles the issue of private information held by the different parties in such computational settings. This paper deals with a complementary problem in such settings: handling the “hidden actions” that are performed by the different parties.

Our model is a combinatorial variant of the classical principal-agent problem from economic theory. In our setting a principal must motivate a team of strategic agents to exert costly effort on his behalf, but their actions are hidden from him. Our focus is on cases where *complex combinations* of the efforts of the agents influence the outcome. The principal motivates the agents by offering to them a set of contracts, which together put the agents in an equilibrium point of the induced game. We present formal models for this setting, suggest and embark on an analysis of some basic issues, but leave many questions open.

Categories and Subject Descriptors

J.4 [Social and Behavioral Sciences]: Economics; K.4.4 [Electronic Commerce]: Payment schemes; C.2.4 [Computer-Communication Networks]: Distributed Systems

General Terms

Design, Economics, Theory

Keywords

Agency Theory, Principal-Agent Model, Incentives, Con-

*A full version of this paper is available at the authors’ web sites. All proofs and some additional results appear in the full version.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC’06, June 11–15, 2006, Ann Arbor, Michigan, USA.
Copyright 2006 ACM 1-59593-236-4/06/0006 ...\$5.00.

tracts, Hidden-Action, Mechanism Design, Moral Hazard, Price of Unaccountability

1. INTRODUCTION

1.1 Background

One of the most striking characteristics of modern computer networks – in particular the Internet – is that different parts of it are owned and operated by different individuals, firms, and organizations. The analysis and design of protocols for this environment thus naturally needs to take into account the different “selfish” economic interests of the different participants. Indeed, the last few years have seen much work addressing this issue using game-theoretic notions (see [7] for an influential survey). A significant part of the difficulty stems from underlying asymmetries of information: one participant may not know everything that is known or done by another. In particular, the field of algorithmic mechanism design [6] uses appropriate incentives to “extract” the private information from the participants.

This paper deals with the complementary lack of knowledge, that of *hidden actions*. In many cases the actual behaviors – actions – of the different participants are “hidden” from others and only influence the final outcome indirectly. “Hidden” here covers a wide range of situations including “not precisely measurable”, “costly to determine”, or even “non-contractible” – meaning that it can not be formally used in a legal contract.

An example that was discussed in [3] is Quality of Service routing in a network: every intermediate link or router may exert a different amount of “effort” (priority, bandwidth, ...) when attempting to forward a packet of information. While the final outcome of whether a packet reached its destination is clearly visible, it is rarely feasible to monitor the exact amount of effort exerted by each intermediate link – how can we ensure that they really do exert the appropriate amount of effort? Many other complex resource allocation problems exhibit similar hidden actions, e.g., a task that runs on a collection of shared servers may be allocated, by each server, an unknown percentage of the CPU’s processing power or of the physical memory. How can we ensure that the right combination of allocations is *actually made* by the different servers? A related class of examples concerns security issues: each “link” in a complex system may exert different levels of effort for protecting some desired security property of the system. How can we ensure that the desired level of

collective security is obtained?

Our approach to this problem is based on the well studied principal-agent problem in economic theory: How can a “principal” motivate a rational “agent” to exert costly effort towards the welfare of the principal? The crux of the model is that the agent’s action (i.e. whether he exerts effort or not) is invisible to the principal and only the final outcome, which is probabilistic and also influenced by other factors, is visible. This problem is well studied in many contexts in classical economic theory and we refer the readers to introductory texts on economic theory such as [5] Chapter 14. The solution is based on the observation that a properly designed contract, in which the payments are contingent upon the final outcome, can influence a rational agent to exert the required effort.

In this paper we initiate a general study of handling *combinations* of agents rather than a single agent. While much work was already done on motivating teams of agents [4], our emphasis is on dealing with the complex combinatorial structure of dependencies between agents’ actions. In the general case, each combination of efforts exerted by the n different agents may result in a different expected gain for the principal. The general question asks which conditional payments should the principal offer to which agents as to maximize his net utility? In our setting and unlike in previous work (see, e.g., [12]), the main challenge is to determine the optimal amount of effort desired from each agent.

This paper suggest models for and provides some interesting initial results about this “combinatorial agency” problem. We believe that we have only scratched the surface and leave many open questions, conjectures, and directions for further research.

We believe that this type of analysis may also find applications in “regular” economic activity. Consider for example a firm that sub-contracts a family of related tasks to many individuals (or other firms). It will often not be possible to exactly monitor the actual effort level of each sub-contractor (e.g., in cases of public-relations activities, consulting activities, or any activities that require cooperation between different sub-contractors.) When the dependencies between the different subtasks are complex, we believe that combinatorial agency models can offer a foundation for the design of contracts with appropriate incentives.

It may also be useful to view our work as part of a general research agenda stemming from the fact that all types of economic activity are increasingly being handled with the aid of sophisticated computer systems. In general, in such computerized settings, complex scenarios involving multiple agents and goods can naturally occur, and they need to be algorithmically handled. This calls for the study of the standard issues in economic theory in new complex settings. The principal-agent problem is a prime example where such complex settings introduce new challenges.

1.2 Our Models

We start by presenting a general model: in this model each of n agents has a set of possible *actions*, the combination of actions by the players results in some *outcome*, where this happens probabilistically. The main part of the specification of a problem in this model is a function that specifies this distribution for each n -tuple of agents’ actions. Additionally, the problem specifies the principal’s utility for each possible outcome, and for each agent, the agent’s cost

for each possible action. The principal motivates the agents by offering to each of them a *contract* that specifies a payment for each possible outcome of the whole project¹. Key here is that the actions of the players are non-observable and thus the contract cannot make the payments directly contingent on the actions of the players, but rather only on the outcome of the whole project.

Given a set of contracts, the agents will each optimize his own utility: i.e. will choose the action that maximizes his expected payment minus the cost of his action. Since the outcome depends on the actions of all players together, the agents are put in a game and are assumed to reach a Nash equilibrium². The principal’s problem, our problem in this paper, is of designing an optimal set of contracts: i.e. contracts that maximize his expected utility from the outcome, minus his expected total payment. The main difficulty is that of determining the required Nash equilibrium point.

In order to focus on the main issues, the rest of the paper deals with the basic binary case: each agent has only two possible actions “exert effort” and “shirk” and there are only two possible outcomes “success” and “failure”. It seems that this case already captures the main interesting ingredients³. In this case, each agent’s problem boils down to whether to exert effort or not, and the principal’s problem boils down to which agents should be contracted to exert effort. This model is still pretty abstract, and every problem description contains a complete table specifying the success probability for each subset of the agents who exert effort.

We then consider a more concrete model which concerns a subclass of problem instances where this exponential size table is succinctly represented. This subclass will provide many natural types of problem instances. In this subclass every agent performs a subtask which succeeds with a low probability γ if the agent does not exert effort and with a higher probability $\delta > \gamma$, if the agent does exert effort. The whole project succeeds as a deterministic Boolean function of the success of the subtasks. This Boolean function can now be represented in various ways. Two basic examples are the “AND” function in which the project succeeds only if all subtasks succeed, and the “OR” function which succeeds if any of the subtasks succeeds. A more complex example considers a communication network, where each agent controls a single edge, and success of the subtask means that a message is forwarded by that edge. “Effort” by the edge increases this success probability. The complete project succeeds if there is a complete path of successful edges between a given source and sink. Complete definitions of the models appear in Section 2.

1.3 Our Results

¹One could think of a different model in which the agents have intrinsic utility from the outcome and payments may not be needed, as in [10, 11].

²In this paper our philosophy is that the principal can “suggest” a Nash equilibrium point to the agents, thus focusing on the “best” Nash equilibrium. One may alternatively study the worst case equilibrium as in [12], or alternatively, attempt modeling some kind of an extensive game between the agents, as in [9, 10, 11].

³However, some of the more advanced questions we ask for this case can be viewed as instances of the general model.

We address a host of questions and prove a large number of results. We believe that despite the large amount of work that appears here, we have only scratched the surface. In many cases we were not able to achieve the general characterization theorems that we desired and had to settle for analyzing special cases or proving partial results. In many cases, simulations reveal structure that we were not able to formally prove. We present here an informal overview of the issues that we studied, what we were able to do, and what we were not. The full treatment of most of our results appears only in the extended version [2], and only some are discussed, often with associated simulation results, in the body of the paper.

Our first object of study is the structure of the class of sets of agents that can be contracted for a given problem instance. Let us fix a given function describing success probabilities, fix the agent’s costs, and let us consider the set of contracted agents for different values of the principal’s associated value from success. For very low values, no agent will be contracted since even a single agent’s cost is higher than the principal’s value. For very high values, all agents will always be contracted since the marginal contribution of an agent multiplied by the principal’s value will overtake any associated payment. What happens for intermediate principal’s values?

We first observe that there is a finite number of “transitions” between different sets, as the principal’s project value increases. These transitions behave very differently for different functions. For example, we show that for the *AND* function only a single transition occurs: for low enough values no agent will be contracted, while for higher values all agents will be contracted – there is no intermediate range for which only some of the agents are contracted. For the *OR* function, the situation is opposite: as the principal’s value increases, the set of contracted agents increases one-by-one. We are able to fully characterize the types of functions for which these two extreme types of transitions behavior occur. However, the structure of these transitions in general seems quite complex, and we were not able to fully analyze them even in simple cases like the “Majority function” (the project succeeds if a majority of subtasks succeeds) or very simple networks. We do have several partial results, including a construction with an exponential number of transitions.

During the previous analysis we also study what we term “the price of unaccountability”: How much is the social utility achieved under the optimal contracts worse than what could be achieved in the non-strategic case⁴, where the socially optimal actions are simply dictated by the principal? We are able to fully analyze this price for the “AND” function, where it is shown to tend to infinity as the number of agents tends to infinity. More general analysis remains an open problem.

Our analysis of these questions sheds light on the difficulty of the various natural associated algorithmic problems. In particular, we observe that the optimal contract can be found in time polynomial in the explicit representation of the probability function. We prove a lower bound that shows that the optimal contract cannot be found in number of queries that is polynomial just in the number of agents, in a general black-box model. We also show that when the probability function is succinctly represented as

⁴The non-strategic case is often referred to as the case with “contractible actions” or the principal’s “first-best” solution.

a read-once network, the problem becomes $\#P$ -hard. The status of some algorithmic questions remains open, in particular that of finding the optimal contract for technologies defined by serial-parallel networks.

In a follow-up paper [1] we deal with equilibria in mixed strategies and show that the principal can gain from inducing a mixed-Nash equilibrium between the agents rather than a pure one. We also show cases where the principal can gain by asking agents to reduce their effort level, even when this effort comes for free. Both phenomena can not occur in the non-strategic setting.

2. MODEL AND PRELIMINARIES

2.1 The General Setting

A principal employs a set of agents N of size n . Each agent $i \in N$ has a possible set of actions A_i , and a cost (effort) $c_i(a_i) \geq 0$ for each possible action $a_i \in A_i$ ($c_i : A_i \rightarrow \mathbb{R}_+$). The actions of all players determine, in a probabilistic way, a “contractible” outcome $o \in O$, according to a success function $t : A_1 \times \dots \times A_n \rightarrow \Delta(O)$ (where $\Delta(O)$ denotes the set of probability distributions on O). A technology is a pair, (t, c) , of a success function, t , and cost functions, $c = (c_1, c_2, \dots, c_n)$. The principal has a certain value for each possible outcome, given by the function $v : O \rightarrow \mathbb{R}$. As we will only consider risk-neutral players in this paper⁵, we will also treat v as a function on $\Delta(O)$, by taking simple expected value. Actions of the players are invisible, but the final outcome o is visible to him and to others (in particular the court), and he may design enforceable contracts based on the final outcome. Thus the contract for agent i is a function (payment) $p_i : O \rightarrow \mathbb{R}$; again, we will also view p_i as a function on $\Delta(O)$.

Given this setting, the agents have been put in a game, where the utility of agent i under the vector of actions $a = (a_1, \dots, a_n)$ is given by $u_i(a) = p_i(t(a)) - c_i(a_i)$. The agents will be assumed to reach Nash equilibrium, if such equilibrium exists. The principal’s problem (which is our problem in this paper) is how to design the contracts p_i as to maximize his own expected utility $u(a) = v(t(a)) - \sum_i p_i(t(a))$, where the actions a_1, \dots, a_n are at Nash-equilibrium. In the case of multiple Nash equilibria we let the principal choose the equilibrium, thus focusing on the “best” Nash equilibrium. A variant, which is similar in spirit to “strong implementation” in mechanism design would be to take the worst Nash equilibrium, or even, stronger yet, to require that only a single equilibrium exists. Finally, the *social welfare* for $a \in A$ is $u(a) + \sum_{i \in N} u_i(a) = v(t(a)) - \sum_{i \in N} c_i(a_i)$.

2.2 The Binary-Outcome Binary-Action Model

We wish to concentrate on the complexities introduced by the combinatorial structure of the success function t , we restrict ourselves to a simpler setting that seems to focus more clearly on the structure of t . A similar model was used in [12]. We first restrict the action spaces to have only two states (binary-action): 0 (low effort) and 1 (high effort). The cost function of agent i is now just a scalar $c_i > 0$ denoting the cost of exerting high effort (where the low effort has cost 0). The vector of costs is $\vec{c} = (c_1, c_2, \dots, c_n)$,

⁵The risk-averse case would obviously be a natural second step in the research of this model, as has been for non-combinatorial scenarios.

and we use the notation (t, \vec{c}) to denote a technology in such a binary-outcome model. We then restrict the outcome space to have only two states (binary-outcome): 0 (project failure) and 1 (project success). The principal's value for a successful project is given by a scalar $v > 0$ (where the value of project failure is 0). We assume that the principal can pay the agents but not fine them (known as the *limited liability* constraint). The contract to agent i is thus now given by a scalar value $p_i \geq 0$ that denotes the payment that i gets in case of project success. If the project fails, the agent gets 0. When the lowest cost action has zero cost (as we assume), this immediately implies that the participation constraint holds.

At this point the success function t becomes a function $t : \{0, 1\}^n \rightarrow [0, 1]$, where $t(a_1, \dots, a_n)$ denotes the probability of project success where players with $a_i = 0$ do not exert effort and incur no cost, and players with $a_i = 1$ do exert effort and incur a cost of c_i .

As we wish to concentrate on motivating agents, rather than on the coordination between agents, we assume that more effort by an agent always leads to a better probability of success, i.e. that the success function t is strictly monotone. Formally, if we denote by $a_{-i} \in A_{-i}$ the $(n-1)$ -dimensional vector of the actions of all agents excluding agent i . i.e., $a_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$, then a success function must satisfy:

$$\forall i \in N, \forall a_{-i} \in A_{-i} \quad t(1, a_{-i}) > t(0, a_{-i})$$

Additionally, we assume that $t(a) > 0$ for any $a \in A$ (or equivalently, $t(0, 0, \dots, 0) > 0$).

DEFINITION 1. *The marginal contribution of agent i , denoted by Δ_i , is the difference between the probability of success when i exerts effort and when he shirks.*

$$\Delta_i(a_{-i}) = t(1, a_{-i}) - t(0, a_{-i})$$

Note that since t is monotone, Δ_i is a strictly positive function. At this point we can already make some simple observations. The best action, $a_i \in A_i$, of agent i can now be easily determined as a function of what the others do, $a_{-i} \in A_{-i}$, and his contract p_i .

CLAIM 1. *Given a profile of actions a_{-i} , agent i 's best strategy is $a_i = 1$ if $p_i \geq \frac{c_i}{\Delta_i(a_{-i})}$, and is $a_i = 0$ if $p_i \leq \frac{c_i}{\Delta_i(a_{-i})}$. (In the case of equality the agent is indifferent between the two alternatives.)*

As $p_i \geq \frac{c_i}{\Delta_i(a_{-i})}$ if and only if $u_i(1, a_{-i}) = p_i \cdot t(1, a_{-i}) - c_i \geq p_i \cdot t(0, a_{-i}) = u_i(0, a_{-i})$, i 's best strategy is to choose $a_i = 1$ in this case.

This allows us to specify the contracts that are the principal's optimal, for inducing a given equilibrium.

OBSERVATION 1. *The best contracts (for the principal) that induce $a \in A$ as an equilibrium are $p_i = 0$ for agent i who exerts no effort ($a_i = 0$), and $p_i = \frac{c_i}{\Delta_i(a_{-i})}$ for agent i who exerts effort ($a_i = 1$).*

In this case, the expected utility of agent i who exerts effort is $c_i \cdot \left(\frac{t(1, a_{-i})}{\Delta_i(a_{-i})} - 1 \right)$, and 0 for an agent who shirk. The principal's expected utility is given by $u(a, v) = (v - P) \cdot t(a)$, where P is the total payment in case of success, given by $P = \sum_{i|a_i=1} \frac{c_i}{\Delta_i(a_{-i})}$.

We say that the principal *contracts with agent i* if $p_i > 0$ (and $a_i = 1$ in the equilibrium $a \in A$). The principal's goal is to maximize his utility given his value v , i.e. to determine the profile of actions $a^* \in A$, which gives the highest value of $u(a, v)$ in equilibrium. Choosing $a \in A$ corresponds to choosing a set S of agents that exert effort ($S = \{i|a_i = 1\}$). We call the set of agents S^* that the principal contracts with in a^* ($S^* = \{i|a_i^* = 1\}$) an *optimal contract* for the principal at value v . We sometimes abuse notation and denote $t(S)$ instead of $t(a)$, when S is exactly the set of agents that exert effort in $a \in A$.

A natural yardstick by which to measure this decision is the non-strategic case, i.e. when the agents need not be motivated but are rather controlled directly by the principal (who also bears their costs). In this case the principal will simply choose the profile $a \in A$ that optimizes the social welfare (global efficiency), $t(a) \cdot v - \sum_{i|a_i=1} c_i$. The worst ratio between the social welfare in this non-strategic case and the social welfare for the profile $a \in A$ chosen by the principal in the agency case, may be termed the *price of unaccountability*.

Given a technology (t, \vec{c}) , let $S^*(v)$ denote the optimal contract in the agency case and let $S_{ns}^*(v)$ denote an optimal contract in the non-strategic case, when the principal's value is v . The social welfare for value v when the set S of agents is contracted is $t(S) \cdot v - \sum_{i \in S} c_i$ (in both the agency and non-strategic cases).

DEFINITION 2. *The price of unaccountability $POU(t, \vec{c})$ of a technology (t, \vec{c}) is defined as the worst ratio (over v) between the total social welfare in the non-strategic case and the agency case:*

$$POU(t, \vec{c}) = \text{Sup}_{v>0} \frac{t(S_{ns}^*(v)) \cdot v - \sum_{i \in S_{ns}^*(v)} c_i}{t(S^*(v)) \cdot v - \sum_{i \in S^*(v)} c_i}$$

In cases where several sets are optimal in the agency case, we take the worst set (i.e., the set that yields the lowest social welfare).

When the technology (t, \vec{c}) is clear in the context we will use POU to denote the price of unaccountability for technology (t, \vec{c}) . Note that the POU is at least 1 for any technology.

As we would like to focus on results that derived from properties of the success function, in most of the paper we will deal with the case where all agents have an *identical cost c* , that is $c_i = c$ for all $i \in N$. We denote a technology (t, \vec{c}) with identical costs by (t, c) . For the simplicity of the presentation, we sometimes use the term *technology function* to refer to the success function of the technology.

2.3 Structured Technology Functions

In order to be more concrete, we will especially focus on technology functions whose structure can be described easily as being derived from independent agent tasks – we call these *structured technology functions*. This subclass will first give us some natural examples of technology function, and will also provide a succinct and natural way to represent the technology functions.

In a structured technology function, each individual succeeds or fails in his own “task” independently. The project's success or failure depends, possibly in a complex way, on the set of successful sub-tasks. Thus we will assume a monotone Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which denotes

whether the project succeeds as a function of the success of the n agents' tasks (and is not determined by any set of $n-1$ agents). Additionally there are constants $0 < \gamma_i < \delta_i < 1$, where γ_i denotes the probability of success for agent i if he does not exert effort, and $\delta_i (> \gamma_i)$ denotes the probability of success if he does exert effort. In order to reduce the number of parameters, we will restrict our attention to the case where $\gamma_1 = \dots = \gamma_n = \gamma$ and $\delta_1 = \dots = \delta_n = 1 - \gamma$ thus leaving ourselves with a single parameter γ s.t. $0 < \gamma < \frac{1}{2}$.

Under this structure, the technology function t is defined by $t(a_1, \dots, a_n)$ being the probability that $f(x_1, \dots, x_n) = 1$ where the bits x_1, \dots, x_n are chosen according to the following distribution: if $a_i = 0$ then $x_i = 1$ with probability γ and $x_i = 0$ with probability $1 - \gamma$; otherwise, i.e. if $a_i = 1$, then $x_i = 1$ with probability $1 - \gamma$ and $x_i = 0$ with probability γ . We denote $x = (x_1, \dots, x_n)$.

The question of the representation of the technology function is now reduced to that of representing the underlying monotone Boolean function f . In the most general case, the function f can be given by a general monotone Boolean circuit. An especially natural sub-class of functions in the structured technologies setting would be functions that can be represented as a *read-once network* – a graph with a given source and sink, where every edge is labeled by a different player. The project succeeds if the edges that belong to player's whose task succeeded form a path between the source and the sink⁶.

A few simple examples should be in order here:

1. The "AND" technology: $f(x_1, \dots, x_n)$ is the logical conjunction of x_i ($f(x) = \bigwedge_{i \in N} x_i$). Thus the project succeeds only if all agents succeed in their tasks. This is shown graphically as a read-once network in Figure 1(a). If m agents exert effort ($\sum_i a_i = m$), then $t(a) = t_m = \gamma^{n-m}(1 - \gamma)^m$. E.g. for two players, the technology function $t(a_1 a_2) = t_{a_1 + a_2}$ is given by $t_0 = t(00) = \gamma^2$, $t_1 = t(01) = t(10) = \gamma(1 - \gamma)$, and $t_2 = t(11) = (1 - \gamma)^2$.
2. The "OR" technology: $f(x_1, \dots, x_n)$ is the logical disjunction of x_i ($f(x) = \bigvee_{i \in N} x_i$). Thus the project succeeds if at least one of the agents succeed in their tasks. This is shown graphically as a read-once network in Figure 1(b). If m agents exert effort, then $t_m = 1 - \gamma^m(1 - \gamma)^{n-m}$. E.g. for two players, the technology function is given by $t(00) = 1 - (1 - \gamma)^2$, $t(01) = t(10) = 1 - \gamma(1 - \gamma)$, and $t(11) = 1 - \gamma^2$.
3. The "Or-of-And's" (OOA) technology: $f(x)$ is the logical disjunction of conjunctions. In the simplest case of equal-length clauses (denote by n_c the number of clauses and by n_l their length), $f(x) = \bigvee_{j=1}^{n_c} (\bigwedge_{k=1}^{n_l} x_k^j)$. Thus the project succeeds if in at least one clause all agents succeed in their tasks. This is shown graphically as a read-once network in Figure 2(a). If m_i agents on path i exert effort, then $t(m_1, \dots, m_{n_c}) = 1 - \prod_i (1 - \gamma^{n_l - m_i} (1 - \gamma)^{m_i})$. E.g. for four players, the technology function $t(a_1^1 a_2^1, a_1^2 a_2^2)$ is given by $t(00, 00) = 1 - (1 - \gamma^2)^2$, $t(01, 00) = t(10, 00) = t(00, 01) = t(00, 10) = 1 - (1 - \gamma(1 - \gamma))(1 - \gamma^2)$, and so on.

⁶One may view this representation as directly corresponding to the project of delivering a message from the source to the sink in a real network of computers, with the edges being controlled by selfish agents.

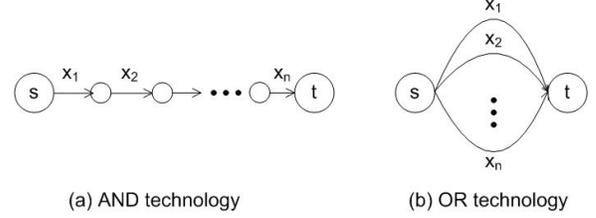


Figure 1: Graphical representations of (a) AND and (b) OR technologies.

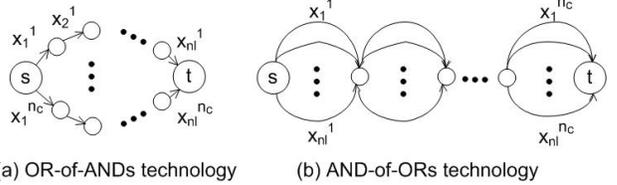


Figure 2: Graphical representations of (a) OOA and (b) AOO technologies.

4. The "And-of-Ors" (AOO) technology: $f(x)$ is the logical conjunction of disjunctions. In the simplest case of equal-length clauses (denote by n_l the number of clauses and by n_c their length), $f(x) = \bigwedge_{j=1}^{n_l} (\bigvee_{k=1}^{n_c} x_k^j)$. Thus the project succeeds if at least one agent from each disjunctive-form-clause succeeds in his tasks. This is shown graphically as a read-once network in Figure 2(b). If m_i agents on clause i exert effort, then $t(m_1, \dots, m_{n_c}) = \prod_i (1 - \gamma^{m_i} (1 - \gamma)^{n_c - m_i})$. E.g. for four players, the technology function $t(a_1^1 a_2^1, a_1^2 a_2^2)$ is given by $t(00, 00) = (1 - (1 - \gamma^2)^2)$, $t(01, 00) = t(10, 00) = t(00, 01) = t(00, 10) = (1 - \gamma(1 - \gamma))(1 - (1 - \gamma)^2)$, and so on.
5. The "Majority" technology: $f(x)$ is "1" if a majority of the values x_i are 1. Thus the project succeeds if most players succeed. The majority function, even on 3 inputs, can not be represented by a read-once network, but is easily represented by a monotone Boolean formula $maj(x, y, z) = xy + yz + xz$. In this case the technology function is given by $t(000) = 3\gamma^2(1 - \gamma) + \gamma^3$, $t(001) = t(010) = t(100) = \gamma^3 + 2(1 - \gamma)^2\gamma + \gamma^2(1 - \gamma)$, etc.

3. ANALYSIS OF SOME ANONYMOUS TECHNOLOGIES

A success function t is called *anonymous* if it is symmetric with respect to the players. I.e. $t(a_1, \dots, a_n)$ depends only on $\sum_{i \in N} a_i$ (the number of agents that exert effort). A technology (t, c) is *anonymous* if t is anonymous and the cost c is identical to all agents. Of the examples presented above, the AND, OR, and majority technologies were anonymous (but not AOO and OOA). As for an anonymous t only the number of agents that exert effort is important, we can shorten the notations and denote $t_m = t(1^m, 0^{n-m})$, $\Delta_m = t_{m+1} - t_m$, $p_m = \frac{c}{\Delta_{m-1}}$ and $u_m = t_m \cdot (v - m \cdot p_m)$, for the case of identical cost c .

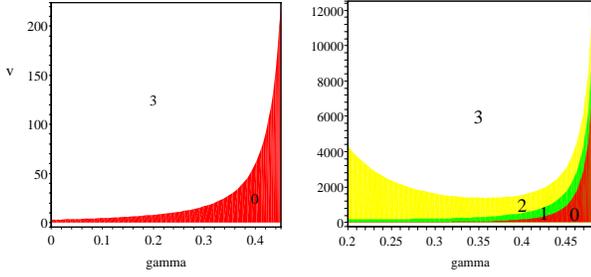


Figure 3: Number of agents in the optimal contract of the *AND* (left) and *OR* (right) technologies with 3 players, as a function of γ and v . *AND* technology: either 0 or 3 agents are contracted, and the transition value is monotonic in γ . *OR* technology: for any γ we can see all transitions.

3.1 *AND* and *OR* Technologies

Let us start with a direct and full analysis of the *AND* and *OR* technologies for two players for the case $\gamma = 1/4$ and $c = 1$.

EXAMPLE 1. *AND* technology with two agents, $c = 1$, $\gamma = 1/4$: we have $t_0 = \gamma^2 = 1/16$, $t_1 = \gamma(1 - \gamma) = 3/16$, and $t_2 = (1 - \gamma)^2 = 9/16$ thus $\Delta_0 = 1/8$ and $\Delta_1 = 3/8$. The principal has 3 possibilities: contracting with 0, 1, or 2 agents. Let us write down the expressions for his utility in these 3 cases:

- **0 Agents:** No agent is paid thus and the principal's utility is $u_0 = t_0 \cdot v = v/16$.
- **1 Agent:** This agent is paid $p_1 = c/\Delta_0 = 8$ on success and the principal's utility is $u_1 = t_1(v - p_1) = 3v/16 - 3/2$.
- **2 Agents:** each agent is paid $p_2 = c/\Delta_1 = 8/3$ on success, and the principal's utility is $u_2 = t_2(v - 2p_2) = 9v/16 - 3$.

Notice that the option of contracting with one agent is always inferior to either contracting with both or with none, and will never be taken by the principal. The principal will contract with no agent when $v < 6$, with both agents whenever $v > 6$, and with either non or both for $v = 6$.

This should be contrasted with the non-strategic case in which the principal completely controls the agents (and bears their costs) and thus simply optimizes globally. In this case the principal will make both agents exert effort whenever $v \geq 4$. Thus for example, for $v = 6$ the globally optimal decision (non-strategic case) would give a global utility of $6 \cdot 9/16 - 2 = 11/8$ while the principal's decision (in the agency case) would give a global utility of $3/8$, giving a ratio of $11/3$.

It turns out that this is the worst price of unaccountability in this example, and it is obtained exactly at the transition point of the agency case, as we show below.

EXAMPLE 2. *OR* technology with two agents, $c = 1$, $\gamma = 1/4$: we have $t_0 = 1 - (1 - \gamma)^2 = 7/16$, $t_1 = 1 - \gamma(1 - \gamma) = 13/16$, and $t_2 = 1 - \gamma^2 = 15/16$ thus $\Delta_0 = 3/8$ and $\Delta_1 = 1/8$. Let us write down the expressions for the principal's utility in these three cases:

- **0 Agents:** No agent is paid and the principal's utility is $u_0 = t_0 \cdot v = 7v/16$.
- **1 Agent:** This agent is paid $p_1 = c/\Delta_0 = 8/3$ on success and the principal's utility is $u_1 = t_1(v - p_1) = 13v/16 - 13/6$.
- **2 Agents:** each agent is paid $p_2 = c/\Delta_1 = 8$ on success, and the principal's utility is $u_2 = t_2(v - 2p_2) = 15v/16 - 15/2$.

Now contracting with one agent is better than contracting with none whenever $v > 52/9$ (and is equivalent for $v = 52/9$), and contracting with both agents is better than contracting with one agent whenever $v > 128/3$ (and is equivalent for $v = 128/3$), thus the principal will contract with no agent for $0 \leq v \leq 52/9$, with one agent for $52/9 \leq v \leq 128/3$, and with both agents for $v \geq 128/3$.

In the non-strategic case, in comparison, the principal will make a single agent exert effort for $v > 8/3$, and the second one exert effort as well when $v > 8$.

It turns out that the price of unaccountability here is $19/13$, and is achieved at $v = 52/9$, which is exactly the transition point from 0 to 1 contracted agents in the agency case. This is not a coincidence that in both the *AND* and *OR* technologies the POU is obtained for v that is a transition point (see full proof in [2]).

LEMMA 1. For any given technology (t, \vec{c}) the price of unaccountability $POU(t, \vec{c})$ is obtained at some value v which is a transition point, of either the agency or the non-strategic cases.

Proof sketch: We look at all transition points in both cases. For any value lower than the first transition point, 0 agents are contracted in both cases, and the social welfare ratio is 1. Similarly, for any value higher than the last transition point, n agents are contracted in both cases, and the social welfare ratio is 1. Thus, we can focus on the interval between the first and last transition points. Between any pair of consecutive points, the social welfare ratio is between two linear functions of v (the optimal contracts are fixed on such a segment). We then show that for each segment, the supremum ratio is obtained at an end point of the segment (a transition point). As there are finitely many such points, the global supremum is obtained at the transition point with the maximal social welfare ratio. \square

We already see a qualitative difference between the *AND* and *OR* technologies (even with 2 agents): in the first case either all agents are contracted or none, while in the second case, for some intermediate range of values v , exactly one agent is contracted. Figure 3 shows the same phenomena for *AND* and *OR* technologies with 3 players.

THEOREM 1. For any anonymous *AND* technology⁷:

- there exists a value⁸ $v_* < \infty$ such that for any $v < v_*$ it is optimal to contract with no agent, for $v > v_*$ it is optimal to contract with all n agents, and for $v = v_*$, both contracts $(0, n)$ are optimal.

⁷*AND* technology with any number of agents n and any γ , and any identical cost c .

⁸ v_* is a function of n, γ, c .

- the price of unaccountability is obtained at the transition point of the agency case, and is

$$POU = \left(\frac{1}{\gamma} - 1\right)^{n-1} + \left(1 - \frac{\gamma}{1-\gamma}\right)$$

Proof sketch: For any fixed number of contracted agents, k , the principal's utility is a linear function in v , where the slope equals the success probability under k contracted agents. Thus, the optimal contract corresponds to the maximum over a set of linear functions. Let v_* denote the point at which the principal is indifferent between contracting with 0 or n agents. In [2] we show that at v_* , the principal's utility from contracting with 0 (or n) agents is higher than his utility when contracting with any number of agents $k \in \{1, \dots, n-1\}$. As the number of contracted agents is monotonic non-decreasing in the value (due to Lemma 3), for any $v < v_*$, contracting with 0 agents is optimal, and for any $v > v_*$, contracting with n agents is optimal. This is true for both the agency and the non-strategic cases.

As in both cases there is a single transition point, the claim about the price of unaccountability for AND technology is proved as a special case of Lemma 2 below. For AND technology $\frac{t_{n-1}}{t_0} = \frac{(1-\gamma)^{n-1} \cdot \gamma}{\gamma^n} = \left(\frac{1}{\gamma} - 1\right)^{n-1}$ and $\frac{t_{n-1}}{t_n} = \frac{(1-\gamma)^{n-1} \cdot \gamma}{(1-\gamma)^n} = \frac{\gamma}{1-\gamma}$, and the expressions for the POU follows. \square

In [2] we present a general characterization of technologies with a single transition in the agency and the non-strategic cases, and provide a full proof of Theorem 1 as a special case.

The property of a single transition occurs in both the agency and the non-strategic cases, where the transition occurs at a smaller value of v in the non-strategic case. Notice that the POU is not bounded across the AND family of technologies (for various n, γ) as $POU \rightarrow \infty$ either if $\gamma \rightarrow 0$ (for any given $n \geq 2$) or $n \rightarrow \infty$ (for any fixed $\gamma \in (0, \frac{1}{2})$).

Next we consider the OR technology and show that it exhibits all n transitions.

THEOREM 2. *For any anonymous OR technology, there exist finite positive values $v_1 < v_2 < \dots < v_n$ such that for any v s.t. $v_k < v < v_{k+1}$, contracting with exactly k agents is optimal (for $v < v_1$, no agent is contracted, and for $v > v_n$, all n agents are contracted). For $v = v_k$, the principal is indifferent between contracting with $k-1$ or k agents.*

Proof sketch: To prove the claim we define v_k to be the value for which the principal is indifferent between contracting with $k-1$ agents, and contracting with k agents. We then show that for any k , $v_k < v_{k+1}$. As the number of contracted agents is monotonic non-decreasing in the value (due to Lemma 3), $v_1 < v_2 < \dots < v_n$ is a sufficient condition for the theorem to hold. \square

The same behavior occurs in both the agency and the non-strategic case. This characterization is a direct corollary of a more general characterization given in [2].

While in the AND technology we were able to fully determine the POU analytically, the OR technology is more difficult to analyze.

OPEN QUESTION 1. *What is the POU for OR with $n > 2$ agents? Is it bounded by a constant for every n ?*

We are only able to determine the POU of the OR technology for the case of two agents [2]. Even for the 2 agents case we already observe a qualitative difference between the POU in the AND and OR technologies.

OBSERVATION 2. *While in the AND technology the POU for $n = 2$ is not bounded from above (for $\gamma \rightarrow 0$), the highest POU in OR technology with two agents is 2 (for $\gamma \rightarrow 0$).*

3.2 What Determines the Transitions?

Theorems 1 and 2 say that both the AND and OR technologies exhibit the same transition behavior (changes of the optimal contract) in the agency and the non-strategic cases. However, this is not true in general. In [2] we provide a full characterization of the sufficient and necessary conditions for general anonymous technologies to have a single transition and all n transitions. We find that the conditions in the agency case are different than the ones in the non-strategic case.

We are able to determine the POU for any anonymous technology that exhibits a single transition in both the agency and the non-strategic cases (see full proof in [2]).

LEMMA 2. *For any anonymous technology that has a single transition in both the agency and the non-strategic cases, the POU is given by:*

$$POU = 1 + \frac{t_{n-1}}{t_0} - \frac{t_{n-1}}{t_n}$$

and it is obtained at the transition point of the agency case.

Proof sketch: Since the payments in the agency case are higher than in the non-strategic case, the transition point in the agency case occurs for a higher value than in the non-strategic case. Thus, there exists a region in which the optimal numbers of contracted agents in the agency and the non-strategic cases are 0 and n , respectively. By Lemma 1 the POU is obtained at a transition point. As the social welfare ratio is decreasing in v in this region, the POU is obtained at the higher value, that is, at the transition point of the agency case. The transition point in the agency case is the point at which the principal is indifferent between contracting with 0 and with n agents, $v_* = \frac{c \cdot n}{t_n - t_0} \cdot \frac{t_n}{t_n - t_{n-1}}$. Substituting the transition point of the agency case into the POU expression yields the required expression.

$$POU = \frac{v_* \cdot t_n - c \cdot n}{v_* \cdot t_0} = 1 + \frac{t_{n-1}}{t_0} - \frac{t_{n-1}}{t_n}$$

\square

3.3 The MAJORITY Technology

The project under the MAJORITY function succeeds if the majority of the agents succeed in their tasks (see Section 2.3). We are unable to characterize the transition behavior of the MAJORITY technology analytically. Figure 4 presents the optimal number of contracted agents as a function of v and γ , for $n = 5$. The phenomena that we observe in this example (and others that we looked at) leads us to the following conjecture.

CONJECTURE 1. *For any Majority technology (any n, γ and c), there exists l , $1 \leq l \leq \lceil n/2 \rceil$ such that the first transition is from 0 to l agents, and then all the remaining $n-l$ transitions exist.*

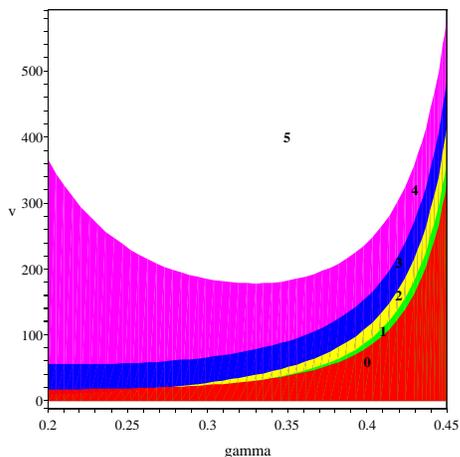


Figure 4: Simulations results showing the number of agents in the optimal contract of the *MAJORITY* technology with 5 players, as a function of γ and v . As γ decreases the first transition is at a lower value and to a higher number of agents. For any sufficiently small γ , the first transition is to $3 = \lceil 5/2 \rceil$ agents, and for any sufficiently large γ , the first transition is to 1 agents. For any γ , the first transition is never to more than 3 agents, and after the first transition we see all following possible transitions.

Moreover, for any fixed c, n , $l = 1$ when γ is close enough to $\frac{1}{2}$, l is a non-decreasing function of γ (with image $\{1, \dots, \lceil n/2 \rceil\}$), and $l = \lceil n/2 \rceil$ when γ is close enough to 0.

4. NON-ANONYMOUS TECHNOLOGIES

In non-anonymous technologies (even with identical costs), we need to talk about the contracted *set* of agents and not only about the number of contracted agents. In this section, we identify the sets of agents that can be obtained as the optimal contract for some v . These sets construct the *orbit* of a technology.

DEFINITION 3. For a technology t , a set of agents S is in the orbit of t if for some value v , the optimal contract is exactly with the set S of agents (where ties between different S 's are broken according to a lexicographic order⁹). The k -orbit of t is the collection of sets of size exactly k in the orbit.

Observe that in the non-strategic case the k -orbit of any technology with identical cost c is of size at most 1 (as all sets of size k has the same cost, only the one with the maximal probability can be on the orbit). Thus, the orbit of any such technology in the non-strategic case is of size at most $n + 1$. We show that the picture in the agency case is very different.

A basic observation is that the orbit of a technology is actually an ordered list of sets of agents, where the order is determined by the following lemma.

LEMMA 3. (Monotonicity lemma) For any technology (t, \vec{c}) , in both the agency and the non-strategic cases, the

⁹This implies that there are no two sets with the same success probability in the orbit.

expected utility of the principal at the optimal contracts, the success probability of the optimal contracts, and the expected payment of the optimal contract, are all monotonically non-decreasing with the value.

PROOF. Suppose the sets of agents S_1 and S_2 are optimal in v_1 and $v_2 < v_1$, respectively. Let $Q(S)$ denote the expected total payment to all agents in S in the case that the principal contracts with the set S and the project succeeds (for the agency case, $Q(S) = t(S) \cdot \sum_{i \in S} \frac{c_i}{t(S) - t(S \setminus i)}$, while for the non-strategic case $Q(S) = \sum_{i \in S} c_i$). The principal's utility is a linear function of the value, $u(S, v) = t(S) \cdot v - Q(S)$. As S_1 is optimal at v_1 , $u(S_1, v_1) \geq u(S_2, v_1)$, and as $t(S_2) \geq 0$ and $v_1 > v_2$, $u(S_2, v_1) \geq u(S_2, v_2)$. We conclude that $u(S_1, v_1) \geq u(S_2, v_2)$, thus the utility is monotonic non-decreasing in the value.

Next we show that the success probability is monotonic non-decreasing in the value. S_1 is optimal at v_1 , thus:

$$t(S_1) \cdot v_1 - Q(S_1) \geq t(S_2) \cdot v_1 - Q(S_2)$$

S_2 is optimal at v_2 , thus:

$$t(S_2) \cdot v_2 - Q(S_2) \geq t(S_1) \cdot v_2 - Q(S_1)$$

Summing these two equations, we get that $(t(S_1) - t(S_2)) \cdot (v_1 - v_2) \geq 0$, which implies that if $v_1 > v_2$ then $t(S_1) \geq t(S_2)$.

Finally we show that the expected payment is monotonic non-decreasing in the value. As S_2 is optimal at v_2 and $t(S_1) \geq t(S_2)$, we observe that:

$$t(S_2) \cdot v_2 - Q(S_2) \geq t(S_1) \cdot v_2 - Q(S_1) \geq t(S_2) \cdot v_2 - Q(S_1)$$

or equivalently, $Q(S_2) \leq Q(S_1)$, which is what we wanted to show. \square

4.1 AOO and OOA Technologies

We begin our discussion of non-anonymous technologies with two examples; the And-of-Ors (*AOO*) and Or-of-Ands (*OOA*) technologies.

The *AOO* technology (see figure 2) is composed of multiple *OR-components* that are “And”ed together.

THEOREM 3. Let h be an anonymous *OR* technology, and let $f = \bigwedge_{j=1}^{n_c} h$ be the *AOO* technology that is obtained by a conjunction of n_c of these *OR-components* on disjoint inputs. Then for any value v , an optimal contract contracts with the same number of agents in each *OR-component*. Thus, the orbit of f is of size at most $n_l + 1$, where n_l is the number of agents in h .

Part of the proof of the theorem (for the complete proof see [2]), is based on such *AOO* technology being a special case of a more general family of technologies, in which disjoint anonymous technologies are “And”-ed together, as explained in the next section. We conjecture that a similar result holds for the *OOA* technology.

CONJECTURE 2. In an *OOA* technology which is a disjunction of the same anonymous paths (with the same number of agents, γ and c , but over disjoint inputs), for any value v the optimal contract is constructed from some number of fully-contracted paths. Moreover, there exist $v_1 < \dots < v_{n_l}$ such that for any v , $v_i \leq v \leq v_{i+1}$, exactly i paths are contracted.

We are unable to prove it in general, but can prove it for the case of an *OOA* technology with two paths of length two (see [2]).

4.2 Orbit Characterization

The *AOO* is an example of a technology whose orbit size is linear in its number of agents. If conjecture 2 is true, the same holds for the *OOA* technology. What can be said about the orbit size of a general non-anonymous technology? In case of identical costs, it is impossible for **all** subsets of agents to be on the orbit. This holds by the observation that the 1-orbit (a single agent that exerts effort) is of size at most 1. Only the agent that gives the highest success probability (when only he exerts effort) can be on the orbit (as he also needs to be paid the least). Nevertheless, we next show that the orbit can have exponential size.

A collection of sets of k elements (out of n) is "admissible", if every two sets in the collection differ by at least 2 elements (e.g. for $k=3$, 123 and 234 can not be together in the collection, but 123 and 345 can be).

THEOREM 4. *Every admissible collection can be obtained as the k – orbit of some t .*

Proof sketch: The proof is constructive. Let \mathcal{S} be some admissible collection of k -size sets. For each set $S \in \mathcal{S}$ in the collection we pick ϵ_S , such that for any two admissible sets $S_i \neq S_j$, $\epsilon_{S_i} \neq \epsilon_{S_j}$. We then define the technology function t as follows: for any $S \in \mathcal{S}$, $t(S) = 1/2 - \epsilon_S$ and $\forall i \in S$, $t(S \setminus i) = 1/2 - 2\epsilon_S$. Thus, the marginal contribution of every $i \in S$ is ϵ_S . Note that since \mathcal{S} is admissible, t is well defined, as for any two sets $S, S' \in \mathcal{S}$ and any two agents i, j , $S \setminus i \neq S' \setminus j$. For any other set Z , we define $t(Z)$ in a way that ensures that the marginal contribution of each agent in Z is a very small ϵ (the technical details appear in the full version). This completes the definition of t .

We show that each admissible set $S \in \mathcal{S}$ is optimal at the value $v_S = \frac{ck}{2\epsilon_S}$. We first show that it is better than any other $S' \in \mathcal{S}$. At the value $v_S = \frac{ck}{2\epsilon_S}$, the set S that corresponds to ϵ_S maximizes the utility of the principal. This result is obtained by taking the derivative of $u(S, v)$. Therefore S yields a higher utility than any other $S' \in \mathcal{S}$. We also pick the range of ϵ_S to ensure that at v_S , S is better than any other set $S' \setminus i$ s.t. $S' \in \mathcal{S}$. Now we are left to show that at v_S , the set S yields a higher utility than any other set $Z \notin \mathcal{S}$. The construction of $t(Z)$ ensures this since the marginal contribution of each agent in Z is such a small ϵ , that the payment is too high for the set to be optimal. \square

In [2] we present the full proof of the theorem, as well as the full proofs of all other claims presented in this section without such a proof. We next show that there exist very large admissible collections.

LEMMA 4. *For any $n \geq k$, there exists an admissible collection of k -size sets of size $\Omega(\frac{1}{n} \cdot \binom{n}{k})$.*

Proof sketch: The proof is based on an error correcting code that corrects one bit. Such a code has a distance ≥ 3 , thus admissible. It is known that there are such codes with $\Omega(2^n/n)$ code words. To ensure that an appropriate fraction of these code words have weight k , we construct a new code by XOR-ing each code word with a random word r . The properties of XOR ensure that the new code remains admissible. Each code word is now uniformly mapped to the whole cube, and thus its probability of having weight k is $\binom{n}{k}/2^n$. Thus the expected number of weight k words is $\Omega(\binom{n}{k}/n)$, and for some r this expectation is achieved or exceeded. \square

For $k = n/2$ we can construct an exponential size admissible collection, which by Theorem 4 can be used to build a technology with exponential size orbit.

COROLLARY 1. *There exists a technology (t, c) with orbit of size $\Omega(\frac{2^n}{n\sqrt{n}})$.*

Thus, we are able to construct a technology with exponential orbit, but this technology is not a network technology or a structured technology.

OPEN QUESTION 2. *Is there a Read Once network with exponential orbit? Is there a structured technology with exponential orbit?*

Nevertheless, so far, we have not seen examples of series-parallel networks whose orbit size is larger than $n + 1$.

OPEN QUESTION 3. *How big can the orbit size of a series-parallel network be?*

We make the first step towards a solution of this question by showing that the size of the orbit of a conjunction of two disjoint networks (taking the two in a serial) is at most the sum of the two networks' orbit sizes.

Let g and h be two Boolean functions on disjoint inputs and let $f = g \wedge h$ (i.e., take their networks in series). The optimal contract for f for some v , denoted by S , is composed of some agents from the h -part and some from the g -part, call them T and R respectively.

LEMMA 5. *Let S be an optimal contract for $f = g \wedge h$ on v . Then, T is an optimal contract for h on $v \cdot t_g(R)$, and R is an optimal contract for g on $v \cdot t_h(T)$.*

Proof sketch: We express the principal's utility $u(S, v)$ from contracting with the set S when his value is v . We abuse notation and use the function to denote the technology as well. Let $\Delta_i^f(S \setminus i)$ denote the marginal contribution of agent $i \in S$. Then, for any $i \in T$, $\Delta_i^f(S \setminus i) = g(R) \cdot \Delta_i^h(T \setminus i)$, and for any $i \in R$, $\Delta_i^f(S \setminus i) = h(T) \cdot \Delta_i^g(R \setminus i)$. By substituting these expressions and $f(S) = h(T) \cdot g(R)$, we derive that $u(S, v) = h(T) \left(g(R) \cdot v - \sum_{i \in T} \frac{c_i}{\Delta_i^h(T \setminus i)} \right) + g(R) \cdot \sum_{i \in R} \frac{c_i}{\Delta_i^g(R \setminus i)}$. The first term is maximized at a set T that is optimal for h on the value $g(R) \cdot v$, while the second term is independent of T and h . Thus, S is optimal for f on v if and only if T is an optimal contract for h on $v \cdot t_g(R)$. Similarly, we show that R is an optimal contract for g on $v \cdot t_h(T)$. \square

LEMMA 6. *The real function $v \rightarrow t_h(T)$, where T is the h – part of an optimal contract for f on v , is monotone non-decreasing (and similarly for the function $v \rightarrow t_g(R)$).*

PROOF. Let $S_1 = T_1 \cup R_1$ be the optimal contract for f on v_1 , and let $S_2 = T_2 \cup R_2$ be the optimal contract for f on $v_2 < v_1$. By Lemma 3 $f(S_1) \geq f(S_2)$, and since $f = g \cdot h$, $f(S_1) = h(T_1) \cdot g(R_1) \geq h(T_2) \cdot g(R_2) = f(S_2)$. Assume in contradiction that $h(T_1) < h(T_2)$, then since $h(T_1) \cdot g(R_1) \geq h(T_2) \cdot g(R_2)$ this implies that $g(R_1) > g(R_2)$. By Lemma 5, T_1 is optimal for h on $v_1 \cdot g(R_1)$, and T_2 is optimal for h on $v_2 \cdot g(R_2)$. As $v_1 > v_2$ and $g(R_1) > g(R_2)$, T_1 is optimal for h on a larger value than T_2 , thus by Lemma 3, $h(T_1) \geq h(T_2)$, a contradiction. \square

Based on Lemma 5 and Lemma 6, we obtain the following Lemma. For the full proof, see [2].

LEMMA 7. *Let g and h be two Boolean functions on disjoint inputs and let $f = g \wedge h$ (i.e., take their networks in series). Suppose x and y are the respective orbit sizes of g and h ; then, the orbit size of f is less or equal to $x + y - 1$.*

By induction we get the following corollary.

COROLLARY 2. *Assume that $\{(g_j, c_j)\}_{j=1}^m$ is a set of anonymous technologies on disjoint inputs, each with identical agent cost (all agents of technology g_j has the same cost c_j). Then the orbit of $f = \bigwedge_{j=1}^m g_j$ is of size at most $(\sum_{j=1}^m n_j) - 1$, where n_j is the number of agents in technology g_j (the orbit is linear in the number of agents).*

In particular, this holds for AOO technology where each OR-component is anonymous.

It would also be interesting to consider a disjunction of two Boolean functions.

OPEN QUESTION 4. *Does Lemma 7 hold also for the Boolean function $f = g \vee h$ (i.e., when the networks g, h are taken in parallel)?*

We conjecture that this is indeed the case, and that the corresponding Lemmas 5 and 7 exist for the OR case as well. If this is true, this will show that series-parallel networks have polynomial size orbit.

5. ALGORITHMIC ASPECTS

Our analysis throughout the paper sheds some light on the algorithmic aspects of computing the best contract. In this section we state these implications (for the proofs see [2]). We first consider the general model where the technology function is given by an arbitrary monotone function t (with rational values), and we then consider the case of structured technologies given by a network representation of the underlying Boolean function.

5.1 Binary-Outcome Binary-Action Technologies

Here we assume that we are given a technology and value v as the input, and our output should be the optimal contract, i.e. the set S^* of agents to be contracted and the contract p_i for each $i \in S^*$. In the general case, the success function t is of size exponential in n , the number of agents, and we will need to deal with that. In the special case of anonymous technologies, the description of t is only the $n+1$ numbers t_0, \dots, t_n , and in this case our analysis in section 3 completely suffices for computing the optimal contract.

PROPOSITION 1. *Given as input the full description of a technology (the values t_0, \dots, t_n and the identical cost c for an anonymous technology, or the value $t(S)$ for all the 2^n possible subsets $S \subseteq N$ of the players, and a vector of costs \vec{c} for non-anonymous technologies), the following can all be computed in polynomial time:*

- *The orbit of the technology in both the agency and the non-strategic cases.*
- *An optimal contract for any given value v , for both the agency and the non-strategic cases.*
- *The price of unaccountability $POU(t, \vec{c})$.*

PROOF. We prove the claims for the non-anonymous case, the proof for the anonymous case is similar.

We first show how to construct the orbit of the technology (the same procedure apply in both cases). To construct the orbit we find all transition points and the sets that are in the orbit. The empty contract is always optimal for $v = 0$. Assume that we have calculated the optimal contracts and the transition points up to some transition point v for which S is an optimal contract with the highest success probability. We show how to calculate the next transition point and the next optimal contract.

By Lemma 3 the next contract on the orbit (for higher values) has a higher success probability (there are no two sets with the same success probability on the orbit). We calculate the next optimal contract by the following procedure. We go over all sets T such that $t(T) > t(S)$, and calculate the value for which the principal is indifferent between contracting with T and contracting with S . The minimal indifference value is the next transition point and the contract that has the minimal indifference value is the next optimal contract. Linearity of the utility in the value and monotonicity of the success probability of the optimal contracts ensure that the above works. Clearly the above calculation is polynomial in the input size.

Once we have the orbit, it is clear that an optimal contract for any given value v can be calculated. We find the largest transition point that is not larger than the value v , and the optimal contract at v is the set with the higher success probability at this transition point.

Finally, as we can calculate the orbit of the technology in both the agency and the non-strategic cases in polynomial time, we can find the price of unaccountability in polynomial time. By Lemma 1 the price of unaccountability $POU(t)$ is obtained at some transition point, so we only need to go over all transition points, and find the one with the maximal social welfare ratio. \square

A more interesting question is whether if given the function t as a black box, we can compute the optimal contract in time that is polynomial in n . We can show that, in general this is not the case:

THEOREM 5. *Given as input a black box for a success function t (when the costs are identical), and a value v , the number of queries that is needed, in the worst case, to find the optimal contract is exponential in n .*

PROOF. Consider the following family of technologies. For some small $\epsilon > 0$ and $k = \lceil n/2 \rceil$ we define the success probability for a given set T as follows. If $|T| < k$, then $t(T) = |T| \cdot \epsilon$. If $|T| > k$, then $t(T) = 1 - (n - |T|) \cdot \epsilon$. For each set of agents \hat{T} of size k , the technology $t_{\hat{T}}$ is defined by $t(\hat{T}) = 1 - (n - |\hat{T}|) \cdot \epsilon$ and $t(T) = |T| \cdot \epsilon$ for any $T \neq \hat{T}$ of size k .

For the value $v = c \cdot (k + 1/2)$, the optimal contract for $t_{\hat{T}}$ is \hat{T} (for the contract \hat{T} the utility of the principal is about $v - c \cdot k = 1/2 \cdot c > 0$, while for any other contract the utility is negative).

If the algorithm queries about at most $\binom{n}{\lceil n/2 \rceil} - 2$ sets of size k , then it cannot always determine the optimal contract (as any of the sets that it has not queried about might be the optimal one). We conclude that $\binom{n}{\lceil n/2 \rceil} - 1$ queries are needed to determine the optimal contract, and this is exponential in n . \square

5.2 Structured Technologies

In this section we will consider the natural representation of read-once networks for the underlying Boolean function. Thus the problem we address will be:

The Optimal Contract Problem for Read Once Networks:

Input: A read-once network $G = (V, E)$, with two specific vertices s, t ; rational values γ_e, δ_e for each player $e \in E$ (and $c_e = 1$), and a rational value v .

Output: A set S of agents who should be contracted in an optimal contract.

Let $t(E)$ denote the probability of success when each edge succeeds with probability δ_e . We first notice that even computing the value $t(E)$ is a hard problem: it is called the network reliability problem and is known to be $\#P$ -hard [8]. Just a little effort will reveal that our problem is not easier:

THEOREM 6. *The Optimal Contract Problem for Read Once Networks is $\#P$ -hard (under Turing reductions).*

PROOF. We will show that an algorithm for this problem can be used to solve the network reliability problem. Given an instance of a network reliability problem $\langle G, \{\zeta_e\}_{e \in E} \rangle$ (where ζ_e denotes e 's probability of success), we define an instance of the optimal contract problem as follows: first define a new graph G' which is obtained by "And"ing G with a new player x , with γ_x very close to $\frac{1}{2}$ and $\delta_x = 1 - \gamma_x$. For the other edges, we let $\delta_e = \zeta_e$ and $\gamma_e = \zeta_e/2$. By choosing γ_x close enough to $\frac{1}{2}$, we can make sure that player x will enter the optimal contract only for very large values of v , after all other agents are contracted (if we can find the optimal contract for any value, it is easy to find a value for which in the original network the optimal contract is E , by keep doubling the value and asking for the optimal contract. Once we find such a value, we choose γ_x s.t. $\frac{c}{1-2\gamma_x}$ is larger than that value). Let us denote $\beta_x = 1 - 2\gamma_x$.

The critical value of v where player x enters the optimal contract of G' , can be found using binary search over the algorithm that supposedly finds the optimal contract for any network and any value. Note that at this critical value v , the principal is indifferent between the set E and $E \cup \{x\}$. Now when we write the expression for this indifference, in terms of $t(E)$ and $\Delta_i^t(E)$, we observe the following.

$$t(E) \cdot \gamma_x \cdot \left(v - \sum_{i \in E} \frac{c}{\gamma_x \cdot \Delta_i^t(E \setminus i)} \right) =$$

$$t(E)(1 - \gamma_x) \left(v - \sum_{i \in E} \frac{c}{(1 - \gamma_x) \cdot \Delta_i^t(E \setminus i)} - \frac{c}{t(E) \cdot \beta_x} \right)$$

if and only if

$$t(E) = \frac{(1 - \gamma_x) \cdot c}{(\beta_x)^2 \cdot v}$$

thus, if we can always find the optimal contract we are also able to compute the value of $t(E)$. \square

In conclusion, computing the optimal contract in general is hard. These results suggest two natural research directions. The first avenue is to study families of technologies

whose optimal contracts can be computed in polynomial time. The second avenue is to explore approximation algorithms for the optimal contract problem.

A possible candidate for the first direction is the family of series-parallel networks, for which the network reliability problem (computing the value of t) is polynomial.

OPEN QUESTION 5. *Can the optimal contract problem for Read Once series-parallel networks be solved in polynomial time?*

We can only handle the non-trivial level of AOO networks:

LEMMA 8. *Given a Read Once AND-of-OR network such that each OR-component is an anonymous technology, the optimal contract problem can be solved in polynomial time.*

Acknowledgments. This work is supported by the Israel Science Foundation, the USA-Israel Binational Science Foundation, the Lady Davis Fellowship Trust, and by a National Science Foundation grant number ANI-0331659.

6. REFERENCES

- [1] M. Babaioff, M. Feldman, and N. Nisan. The Price of Purity and Free-Labor in Combinatorial Agency. In *Working Paper*, 2005.
- [2] M. Babaioff, M. Feldman, and N. Nisan. Combinatorial agency, 2006. www.sims.berkeley.edu/~moshe/comb-agency.pdf.
- [3] M. Feldman, J. Chuang, I. Stoica, and S. Shenker. Hidden-action in multi-hop routing. In *EC'05*, pages 117–126, 2005.
- [4] B. Holmstrom. Moral Hazard in Teams. *Bell Journal of Economics*, 13:324–340, 1982.
- [5] A. Mass-Colell, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [6] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behaviour*, 35:166 – 196, 2001. A preliminary version appeared in STOC 1999.
- [7] C. Papadimitriou. Algorithms, Games, and the Internet. In *Proceedings of 33rd STOC*, pages 749–753, 2001.
- [8] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983.
- [9] A. Ronen and L. Wahrmann. Prediction Games. *WINE*, pages 129–140, 2005.
- [10] R. Smorodinsky and M. Tennenholtz. Sequential Information Elicitation in Multi-Agent Systems. *20th Conference on Uncertainty in AI*, 2004.
- [11] R. Smorodinsky and M. Tennenholtz. Overcoming Free-Riding in Multi-Party Computations - The Anonymous Case. *Forthcoming, GEB*, 2005.
- [12] E. Winter. Incentives and Discrimination. *American Economic Review*, 94:764–773, 2004.