Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

ELSEVIER

Games and Economic Behavior 55 (2006) 385–406

GAMES and Economic Behavior

www.elsevier.com/locate/geb

Overcoming free riding in multi-party computations—The anonymous case [☆]

Rann Smorodinsky [*], Moshe Tennenholtz

Davidson Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel

Received 22 September 2003

Available online 1 August 2005

Abstract

This paper addresses the question of multi-party computation in a model with asymmetric information. Each agent has a private value (secret), but in contrast to standard models, the agent incurs a cost when retrieving the secret. There is a social choice function the agents would like to compute and implement. All agents would like to perform a joint computation, which input is their vector of secrets. However, agents would like to free-ride on others' contribution.

A mechanism which elicits players' secrets and performs the desired computation defines a game. A mechanism is 'appropriate' if it (weakly) implements the social choice function for all secret vectors. namely, if there exists an equilibrium in which it is able to elicit (sufficiently many) agents' secrets and perform the computation, for all possible secret vectors. We show that 'appropriate' mechanisms approach agents sequentially and that they have low communication complexity.

JEL classification: C72; C80; D70; D82

Keywords: Multi-party computations; Sequential mechanisms; Equilibrium; Revelation principle; Information acquisition

[☆] Both authors acknowledge the financial support of the Technion V.P.R. fund and the Davidson fund. M. Tennenholtz acknowledges the financial support of the ISF Grant 890015.

[*] Corresponding author.

E-mail addresses: rann@ie.technion.ac.il (R. Smorodinsky), moshet@ie.technion.ac.il (M. Tennenholtz).

0899-8256/$ – see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.geb.2005.05.001

1. Introduction

It is well known that group activities, where a joint decision must be reached, may result in inefficient outcomes due to the free riding phenomena. Consider a very simple model where all agents utility, derived from the group decision, are fully aligned. Additionally, they may have (dis-)utility from contributing to the group decision process. We investigate this model in a positive framework and ask how can one overcome the free-riding problem and achieve efficiency

Consider the following motivating setting. Strategic decisions taken by firms, such as a company's product direction or its geographic expansion, are often a result of a management process, and involve an analysis of the various aspects of the organization (R&D, human resources, financial considerations, sales prospects, market analysis and more). Each analysis is undertaken by a different management member, according to her expertise, and the final decision is optimally based on a complete set of recommendations. Often, all the management team is incentivized such that their goals are aligned (e.g., in many high tech firms the management team holds stock options) and so all of them would like to ensure the 'right' decision is reached. However, it may be costly for some (or all) members to do a complete analysis of their domain expertise and this may result in some of the team members relying on others to do a competent analysis, while they suffice in making a recommendation that is not well founded.

In analogy to the above setting, an academic department often wants to recruit high quality candidates, and all department members may have a similar interest. In such recruiting processes we trust each other to evaluate each candidate. Often such an evaluation involves a tedious reading of some of the candidates' work, and consequently a detailed reading is replaced with a glance at the paper.

We treat situations as above in a game-theoretic setting which we refer to as *multi-party computation games*. In a multi-party computation game a society of n agents (players) has access to some private information (secret). Each agent may access his (and only his) secret, but by doing so he incurs a cost. The agents would like to reach a decision, by computing, jointly, a function whose input is the vector of n secrets. Assume agents are selfish and are driven by utility maximization considerations. Additionally, assume all agents' utilities are aligned. Namely, except for the cost of accessing the secret they all have the same utility function. The question we pose is how can one guarantee efficiency in such a setting? This general setting is often an important component in the analysis of problems and models in computer science (e.g., computation based on a distributed data base), statistics (e.g., computing a population statistic such as an order statistic), economics (e.g., computing a clearing price in a two-sided auction), political sciences (e.g., voting) and more. In the various settings one may interpret the access costs as participation costs, computational costs, cost of accessing a data base, revelation costs (due to the revelation of the individuals private opinion/secret) or even a learning cost, depending on the specific model.

As our setting is motivated by models in computer science and in economics we alternate terminology between 'multi-party computations' and 'social-choice functions.'

An example of a multi-party computation game, in the context of distributed computing, is as follows: A classical problem in distributed computing is the need to reach a consensus on the value of a shared variable where the value of this variable might have been corrupted due to machine failure. This value will typically depend on the history of information available to a set of processors. Each processor will therefore need to access its memory and extract from it the relevant information, to be shared with other processors, in order to reach a decision on the value of the shared variable. However, if the processes represent different entities (say, in a distributed operating systems context) then each processor may be interested in free-riding on other processors computations, leading to phenomena as above.

Note that in the proposed setting, accessing one's own private information becomes a strategic question. This approach generates a natural tension between the socially (and even privately) optimal action, which is to compute the joint function correctly, and agents' incentive to free-ride. In order to overcome this tension one may need to design a mechanism to prevent (some or all) agents from free-riding, elicit agents' secrets and execute the desired computation:

Example 1. Assume a candidate to the economics department has already written 11 research papers, and the economics department would like to decide on whether to make her a job offer based on the quality of the papers. There are 11 agents (committee members) who are each given one paper to read in order to make a recommendation. Initially, each paper may be "good" or "bad" with equal probabilities, and the department has chosen to make an offer to a candidate if he has a majority of "good" papers. Consider a situation where a committee member values the correct recommendation of the committee, at 1000 USD to him, but values the time he needs to spend on reading the paper at 400 USD. A simple mechanism asks all the agents, simultaneously, for their secrets. The strategy tuple where all agents choose to read the papers and report truthfully is not an equilibrium. To see this consider the perspective of agent 1. Assuming all agents report (truthfully, or not), then agent 1 can alter the outcome of G only if the other 10 replies split evenly between 0 and 1, which has a probability of $\frac{10!}{5!5!}0.5^{10} \approx 0.25$. Therefore, by guessing, and assuming all other agents compute, he will gain $0.25 \times 500 + 0.75 \times 1000 = 875$ USD. However, by computing an agent gains at most $1000 - 400 = 600$ USD, and so player 1 has no incentive to compute. The same argument holds for all 11 agents.

Example 2. Consider 4 agents, each having a secret of either 0 or 1, drawn independently with equal probabilities. The agents attribute a utility of 1 for implementing the right decision, which simply depends on whether all agents are in consensus or not. Assume three of them have very low computation costs (say, zero) and the fourth has a cost of 0.4. A mechanism that approaches all agents simultaneously may fail to compute correctly as the fourth agent will choose to 'guess' his secret (by guessing his payoff is 0.875, whereas by accessing the secret the payoff is 0.6). However, if the mechanism approaches the first three then the mechanism will either learn the true value (which is the case if there is lack of consensus already among these three) or he may update the fourth agent about the previous replies. In this case the fourth agent's 'pivotalness' increases and so by guessing he expects a payoff of 0.5, compared to 0.6, when not guessing. Consequently, the fourth agent will access his secret and the mechanism will surely compute correctly.

What the last example demonstrates is that the naive way of approaching agents simultaneously will not necessarily work, while a sequential approach will. The intuition behind this example and the advantage of sequential mechanisms is as follows. Assume there are agents with low (computation) costs. These agents can be approached first. It is possible that based on their replies the desired multi party computation can be carried out. However, if this is not the case then, intuitively, the impact of the other agents on the result of the computation (their pivotalness) increases, and so the incentive to incur the cost and compute increases. In other words, as agents are approached sequentially the remaining agents become either redundant for the desired computation or highly pivotal.

The class of multi party computations (social choice functions) we study is that of anonymous functions. An anonymous function is one where the function's value does not depend on the identity of the agents but on the secrets only. In other words, a permutation of agents' secrets will not change the value of the function. This class of functions is quite elementary and often used in models. Among the anonymous functions are majority, consensus, unanimity, average, variance, order statistic, percentile and more.

The basic new terminology we introduce for sequential mechanisms is that of an 'appropriate mechanism.' A mechanism is deemed appropriate if it (weakly) implements, in equilibrium, the social choice function, for all realizations of secrets. As we will see it is often the case that we can only guarantee weak implementation.

The class of sequential mechanisms is quite rich. However, we show that appropriate mechanisms have a simple characterization. More surprisingly, the characterization is agnostic to the structure of the function, as long as this function is anonymous.

Informally, the main results we report are as follows:

- A general chracterization/reduction theorem (Theorems 1 and 2)—In looking for an appropriate mechanism one may reduce the search domain: an appropriate mechanism exists if and only if there exists an appropriate mechanism that is "simple" (in a sense made precise in the paper). The reduction in size of the search domain is quite significant and, may turn an impractical search problem into a practical one. A numerical example follows Theorem 2.
- In some scenarios we can actually further characterize the appropriate mechanisms:
  - If all communication is done via broadcasting, then it is sufficient to look at a kind of "greedy" mechanism.
  - If all agents' costs are equal then it is sufficient look at a mechanism that approaches agents simultaneously.
  - If the mechanism must commit to the order at which the agents are approached, then it is sufficient to look at a mechanism that orders agents by costs—from low to high.

The model we study connects to various strands of the literature. The closest one is the very recent literature on 'non-cooperative computation,' introduced by Shoham and Tennenholtz (2002). In this literature the multi party computation game asserts that agents would like to know the result of the computation but would rather others not learn the result. We briefly discuss here some other branches of the literature which are related to our work, and refer the reader to Section 4 where we elaborate.

Secure multi party computation involves a similar setup with a different incentive scheme. Agents, holding a private secret, would like to compute some value, without disclosing anything about their own secret (at least nothing beyond what is disclosed by the value of the function).

The literature on *Public Goods* studies games and mechanisms where there is a basic tension between agents' individual efforts (contributions) and the desire to supply a public good. Often, this literature demonstrates the impossibility of supplying the public good, even in situations where it is the socially optimal action. In this literature, agents' main consideration is how much they can influence the result of the computation, namely their *pivotalness* versus their expected contribution. The topic of pivotalness has been studied extensively by economists as well as by computer scientists.

The literature on *mechanism design* discusses the existence and characterization of 'optimal' mechanisms, in the same vein that we do. However, the knowledge of private information in those models is assumed. We consider it part of a player's strategy. One should note that, consequently, we cannot rely on the revelation principle, as agents cannot 'reveal' what they do not know.

The next section provides a general model and a detailed definition of 'sequential mechanisms' and Section 3 states all the results. Section 4 discusses related literature and Section 5 provides some natural directions for future research.

2. Model

Let $N = \{1, \dots, n\}$ be a finite set of agents. Each agent j has a unique secret $s_j \in \{0, 1\}$ that he may compute. Let $q \in [0.5, 1)$ be the prior probability of $s_j = 1$ and assume these events are independent. The assumption that $q \in [0.5, 1)$ is without loss of generality (if not, then rename the secrets). Agents may compute their own secrets, however, computation is costly and agent j pays $c_j > 0$ for computing s_j. Without loss of generality we shall assume $c_1 \leqslant c_2 \leqslant \cdots \leqslant c_n$ (in words, agents are ordered by their costs).

Agents are interested in computing some joint binary parameter, i.e. implementing a corresponding social choice function (e.g., the majority vote or whether they have a consensus) that depends on the vector of private inputs. Let $G : \{0, 1\}^n \to \{0, 1\}$ denote the desired computation. Each agent, j, has a utility of v_j from implementing the correct value of G, and a utility of zero for incorrect implementation. We will assume that $v_j > c_j$, otherwise the agent faces no dilemma (we assume no side payments).

In the exposition we will use the convention that $v_j = 1$. This is done without loss of generality, as the more general case where $v_j > c_j > 0$, is equivalent to the case where the value of agent j is 1 but the cost is c_j / v_j.

A central designer elicits the agents' secrets, computes G and reports the computed value of G back to each agent. In this setup each agent faces a dilemma of whether to compute his private secret s_j, at a cost of c_j, or perhaps to submit a guess to the central designer. The desired property of a mechanism is the correct computation of G, which is done through the elicitation of secrets from sufficiently many agents.

In the introduction we considered an example where computation and truth revealing is not an equilibrium. The following example illustrates another situation.

Example 3. Let G be the parity function, $q = 0.5$, and $c_j = 0.4$ for all $j = 1, \dots, 11$. Once again, consider a simple mechanism that asks all the agents simultaneously for their secrets and computes G. In this example each agent is pivotal and therefore all agents computing is an equilibrium, and so G will be computed correctly.

2.1. Sequential mechanisms

Interestingly enough, there may be a strict advantage in approaching agents sequentially. The intuition is that agents with a high computational cost may not be willing to compute, unless convinced they are pivotal. If some agents with low computation cost have already provided their secret the other agents may face one of two situations. Either G can be computed from previous replies or it cannot. In the latter case the remaining agents are more pivotal, perhaps sufficiently so to justify a costly computation.

We now model mechanisms that may approach players sequentially. We begin with the most general construction and later look at natural restrictions.

A *sequential mechanism* (or mechanism) is an ordering of the set of agents, where the kth agent in the order is selected according to the reply of its predecessors. Furthermore the kth agent is provided with some information based on the replies of its predecessors.

Let $\Omega = \{0, 1\}^n$ be the state space of agents' replies. For each subset of indices $M \subset N$ we abuse notation and use $\omega_M \in \{0, 1\}^M$, also to denote the cylinder subset $\{(\omega_M, \omega_{-M})\colon \omega_{-M} \in \{0, 1\}^{-M}\} \subset \{0, 1\}^n$. Let H_M denote the set of all such cylinders, with $H_\emptyset = \emptyset$ (H_M can be viewed as the set of all histories that can be generated after approaching the agents in $M \subset N$). Let F_M denote the field generated by all elements in H_M and finally $H = \bigcup_{M \subset N} H_M$ (think of H as the set of all possible histories).

Definition 1. A *sequential mechanism*, $\mathcal{A} = (\sigma, A)$, is composed of two components. For each $1 \leqslant j \leqslant n$:

(1) The order function, $\sigma : H \to N$, which determines the agent that is approached following any history. For each $M \subset N$ and $h_M \in H_M$ we apply the restriction that $\sigma(h_M) \notin M$.
(2) The information function, $A : H \to 2^H$. For each $h \in H$, the mechanism provides the information $A(h)$ to $\sigma(h)$.

We will denote by σ_j the (random) agent in the jth place in the order induced by σ, and A_j denotes the information provided to agent σ_j, the jth agent. Note that agent σ_j will not necessarily know he is the jth agent. The (random) order of agent j is denoted $\sigma^{-1}(j)$.

We consider some natural restrictions of the above definition:

- A *fixed order* sequential mechanism is one where all the functions σ_i are constant. In words, the order of the agents is set in advance, and is independent of their replies.
- A *truth revealing* sequential mechanism satisfies $h \in A(h)$ for all $h \in H$.
- A sequential mechanism is *fully revealing* if $A(h) = h$ for all $h \in H$ (in which case it must also be truth revealing). This is a natural restriction in situations where all communication is done via broadcast channels (see Section 3.3 for further discussion).

Note that the above definition excludes the possibility of approaching agents more than once. Some examples for various types of mechanisms are:

Example 4. The *simultaneous* mechanism is one where all agents are asked simultaneously. Formally, $\sigma_j = j$ and $A_j = H\ \forall j$. Note that the simultaneous mechanism has a fixed order and is truth revealing. However, it is not fully revealing.

Example 5. Let G be the simple majority function (i.e., $G = 1$ if the number of secrets with value one is greater than or equals to the number of secrets with value zero). Set $\sigma_1 = 1$ and assume that after receiving replies from j agents the algorithm for approaching the next agent is as follows: if the conditional probability for $G = 1$ is greater than or equal to 0.5 then the agent with the lowest index (among the ones who have not been approached yet) is approached, otherwise the agent with the highest index among them is approached. The information given to that agent is the number of secrets of both types received so far. Assuming $q = 0.5$ then this mechanism does not have a fixed order and is not fully revealing. However, it is truth revealing.

Example 6. Let G be an arbitrary function and assume the mechanism approaches the agents according to some fixed order. When approaching agent σ_{j+1} it truthfully tells him whether he is able to compute the value of G from the input so far, or not. This mechanism has a fixed order but is not necessarily fully revealing (perhaps it is easy to think of the consensus function, G, defined by $G(s) = 1$ if and only if $s_1 = s_2 = \cdots = s_n$).

In the sequel, we refer to mechanisms of this type as *Yes/No mechanisms*, and such mechanisms will prove to be central to our analysis.

2.2. Agents' strategies and equilibrium

The action space of each agent, j, is the set {compute, don't compute} × $\{0, 1\}$. The first coordinate refers to whether the agent chooses to go through the costly computation and the second coordinate is what the agent chooses to inform the central mechanism.

Note that this implies that each agent has 6 (and not 4) actions: Don't compute and report 0, Don't compute and report 1, Compute and report 0, Compute and report 1, Compute and report the true computed value, and Compute and report a false value. Let us denote by Γ the set of actions.

A pure strategy for player j, $x_j : 2^H \to \Gamma$, assigns an action to each possible subset of histories, and a (mixed) strategy, $X_j : 2^H \to \Delta(\Gamma)$, assigns a probability distribution over Γ. The parameter q, alongside the tuple of (mixed) strategies, $\{X_j\}_{j=1}^{n}$, determines the probability that G will be computed.

An equilibrium for the mechanism $\mathcal{A}$, is a vector of n strategies, one for each agent, such that each agent's strategy is the best response against the other agents' strategies.

Our first result suggests that for any arbitrary mechanism there exists another mechanism which has the same incentive structure as the original one, and furthermore has two notable properties: it is truth revealing and has the 'partition property,' which is defined below.

Theorem 1. *For any $q \in [0.5, 1)$, mechanism $\mathcal{A} = (\sigma, A)$, and equilibrium $\{X_j\}_{j=1}^n$, of $\mathcal{A}$, there exists a truth revealing mechanism, $\mathcal{B} = (\sigma, B)$ (note it has the same order as $\mathcal{A}$), and an equilibrium, $\{X'_j\}_{j=1}^n$ of $\mathcal{B}$, such that when agent j computes s_j for a given history of secrets in the equilibrium of $\mathcal{A}$ then it also does so in the equilibrium of $\mathcal{B}$. Furthermore, the sets $B_j^1, B_j^2, \ldots, B_j^t$, which are the possible sets reported to agent j in $\mathcal{B}$, are pairwise disjoint (we refer to this as the 'partition property').*

Proof. For each $h \in H$ let $B(h) = \{\hat{h} \in H \mid A(\hat{h}) = A(h)\}$ and set $\mathcal{B} = (\sigma, B)$. It is straightforward to see that $\mathcal{B}$ has the same incentive structure as $\mathcal{A}$, is truth revealing and has the partition property. □

Theorem 1 provides an observation in the spirit of the revelation-principle (e.g., Myerson, 1982). Note that, whereas the proof of the classical revelation principle hinges on the mechanism simulating the agents actions in equilibrium, in the above proof agents actually simulate the mechanism.

Assuming the mechanism is truth revealing then some of the actions are dominated by others:

- Don't compute and report 0 is dominated by Don't compute and report 1 (recall $q = \text{Prob}(s_j = 1) \geqslant 0.5$).
- Compute and report 0, Compute and report 1 and Compute and report a false value are all dominated by Compute and report the true computed value.

So Theorem 1 suggests that we can replace an arbitrary mechanism with one in which agents actions are restricted to the set 'Don't compute and report 1' and 'Compute and report the true computed value.' From now on we will refer to these as 'Guess' and 'Compute.' Formally, for each j, $X_j : 2^H \to \Delta(\{\text{compute, guess}\})$.

2.3. Appropriate mechanisms

We seek mechanisms which can compute the true value of G in equilibrium. In fact, it is required that a mechanism computes G with certainty. Therefore we seek mechanisms that induce sufficiently many agents to compute their true secret, in order for G to be computed. Note that in many cases G may be computed with partial information. For example, in the case of a consensus function it is sufficient to elicit information sequentially until we get two different replies, which are truthful.

Definition 2. A mechanism $\mathcal{A}$ is *appropriate* for G, at $q \in [0.5, 1)$, if there exists an equilibrium where G can surely be computed for all vector of agents' secrets. Such an equilibrium is referred to as a *computing equilibrium*.

Example 7. Let $n = 5$, G be the majority vote, and $c_j = 1/8\ \forall j$. The simultaneous mechanism is appropriate at $q = 0.5$. To see this note that for all agents to compute s_j is an equilibrium. However, for $q = 0.8$ the simultaneous mechanism is not appropriate as in equilibrium no agent computes his secret and so G could be false with positive probability.

Our first result about appropriate mechanisms is a direct corollary of Theorem 1:

Corollary 1. *For any function G, $q \in [0.5, 1)$, and mechanism $\mathcal{A} = (\sigma, A)$ that is appropriate at q, there exists a truth revealing mechanism, $\mathcal{B} = (\sigma, B)$ which has the 'partition property,' and is appropriate at q.*

3. Results for the case of anonymous functions

A function that satisfies $G(s) = G(\sigma(s))$ for any s and any permutation $\sigma : N \to N$ is called *anonymous*. In this paper we limit the discussion to anonymous functions.

First we look at some general results and then study some restricted cases, such as fixed-order mechanisms, equal costs and the case of communication via broadcast channels.

A subset, $h_M \subset H_M$, is called *coherent* if G is constant over all elements of h_M. Otherwise it is incoherent. A collection of subsets, $\{h_{M_1}, \ldots, h_{M_k}\} \in 2^H$ is called *coherent* if for all i, h_{M_i} is coherent. In a coherent collection the value of G over the different subsets may be different. Such a collection is called *incoherent* if *all* the cylinders are incoherent. Note that a collection that is not coherent is not necessarily incoherent.

The next lemma demonstrates the strength of an appropriate mechanism. It claims that in the computing equilibrium of an appropriate mechanism an agent chooses the action 'Guess,' with positive probability, if and only if the information it receives is coherent.

Lemma 1. *Assume G is anonymous and let $\mathcal{A}$ be an arbitrary truth revealing mechanism with the partition property (see Theorem 1). If $\mathcal{A}$ is appropriate for G at q then agent σ_j's action is 'Guess' if and only if A_j is coherent.*

Proof. The first direction is straightforward. If A_j is coherent then G can already be computed from previous replies and so Agent j has no incentive to compute.

What about the converse? As $\mathcal{A}$ is appropriate the value of G can be computed for any vector of signals. Thus, if some agent, σ_j, chooses 'Guess' as his best reply, it must be the case that his signal will not impact the value of G.

Note that as G is anonymous its value can be inferred from the number of coordinates with value zero in the vector of signals. Denote the number of zeroes known to $\mathcal{A}$ at time j by r. Assume, by way of contradiction, that G cannot be computed at stage j. Therefore, there must be some number $n \geqslant t > r$ such that if the total number of zeroes is t then $G = 0$ and if the total number of zeroes is $t + 1$ then $G = 1$ (or the other way around). But this suggests that it is possible that for some sequence of signals agent σ_j will be pivotal, possibly shifting the number of zeroes between t and $t + 1$. Consequently, it will influence the value of G. The conclusion is, therefore, that if agent σ_j guesses it must be the case that G can be computed.

As G is truth revealing this conclusion must hold whenever agent σ_j is provided with the information A_j. Namely for each h such that $A(h) = A_j$ we know that G can be computed, which means that all such h are coherent. As $A_j = \{h \mid A(h) = A_j\}$ we are done. □

We say that a mechanism has the *Yes/No property* if $\forall j,\ A_j$ is either coherent or incoherent.

A *Yes/No mechanism* is a mechanism such that each agent j receives, when it is his turn, one of only two subsets of H, one which is coherent and the other is incoherent. In particular, a Yes/No mechanism has the Yes/No property.

Lemma 2. *Assume G is anonymous and $\mathcal{A}$ is a truth revealing mechanism with the partition property that is appropriate at q. Then there exists a truth revealing mechanism, $\mathcal{B}$, with the partition property, that is appropriate at q and has the Yes/No property. Furthermore, $\mathcal{B}$ has the same* (*random*) *order as $\mathcal{A}$*

Let $Z_j \subset \{0,1\}^n$ denote the set of secrets where agent j is pivotal. Namely $Z_j = \{s \mid G(s) \neq G(s_{-j}, 1 - s_j)\}$. Z_j^c is its complement. When $\mathcal{A}$ is a truth revealing mechanism with the partition property then we can use Z_j to compute the expected utility agent j receives by guessing, at a computing equilibrium. Formally, when given the information $A_j \subset H$, the expected utility from guessing is $\operatorname{Prob}(Z_j \mid A_j) \cdot q + \operatorname{Prob}(Z_j^c \mid A_j) \cdot 1$, in a computing equilibrium. (To see this note that if j is pivotal then it will learn the correct value of G with probability q, otherwise it will learn the value of G with probability 1.)

Proof. We prove this lemma by showing, first, a mechanism where the last agent receives information which is either coherent or incoherent. We then claim that the same argument can work for the agent which is one before the last and so on and so forth.

Consider an arbitrary realization of secrets, s, and consider, σ_n, the last agent to be approached. A_{σ_n} is the information provided to the last agent.

Assume that A_{σ_n} is not coherent. By the previous lemma we know that σ_n's action was to compute. By computing, agent σ_n will get a utility which is no less than the utility from deviating. Therefore $1 - c_{\sigma_n} \geqslant \operatorname{Prob}(Z_{\sigma_n} \mid A_{\sigma_n}) \cdot q + \operatorname{Prob}(Z_{\sigma_n}^c \mid A_{\sigma_n}) \cdot 1$.

Consider an alternative mechanism $\mathcal{B}$ which is equal to $\mathcal{A}$, except that on s it replaces A_{σ_n} with the subset composed of all cylinders that are coherent, denoted $A_{\sigma_n}^{\text{coherent}} \subset A_{\sigma_n}$, or by $A_{\sigma_n}^{\text{incoherent}} = A_{\sigma_n} - A_{\sigma_n}^{\text{coherent}}$, depending on which is true.

The new mechanism has a computing equilibrium, with the following strategies: All agents use the same strategies as in $\mathcal{A}$, except for agent σ_n. For all subsets of 2^H, but $A_{\sigma_n}^{\text{coherent}}$ and $A_{\sigma_n}^{\text{incoherent}}$, this agent acts the same as in $\mathcal{A}$. However in reply to $A_{\sigma_n}^{\text{coherent}}$, this agent guesses, as G can already be computed. In reply to $A_{\sigma_n}^{\text{incoherent}}$ this agent computes. This is actually its best reply because $\operatorname{Prob}(Z_{\sigma_n} \mid A_{\sigma_n}) \leqslant \operatorname{Prob}(Z_{\sigma_n} \mid A_{\sigma_n}^{\text{incoherent}})$, and so σ_n's conditional probability of being pivotal increases (in fact it becomes one). Consequently, the expected payoff from guessing diminishes, compared to A_{σ_n}.

Now we re-visit the other agents' strategies. Their consideration of the probability of being pivotal has not changed in $\mathcal{B}$ and therefore the strategies in the computing equilibrium of $\mathcal{A}$ are best replies in $\mathcal{B}$.

Now repeat this argument, sequentially, for any s, for which A_{σ_n} is not coherent. The resulting mechanism will be such that the (random) last agent will get information which is either coherent or incoherent.

Now, we can use the same arguments for the agent σ_{n-1}, and then to σ_{n-2} and so on until we get the desired mechanism. □

Lemma 3. *Assume G is anonymous and $\mathcal{A}$ is a truth revealing mechanism that is appropriate at q and has the partition property and the Yes/No property. Then there exists a truth revealing Yes/No mechanism, $\mathcal{B}$, that is appropriate at q.*

Proof. First, let us construct the mechanism $\mathcal{B}$. The random order on $\mathcal{B}$ is that of $\mathcal{A}$. For each agent j let B_j denote the union of all coherent subsets of the form $A_{\sigma^{-1}(j)}$. Let B_j^c be its complement. As $\mathcal{A}$ has the Yes/No property, B_j^c must be incoherent and so $\mathcal{B}$ is a Yes/No mechanism.

Assume $\mathcal{A}$ is appropriate for q and let s be an arbitrary vector of secrets. Denote by σ_n the last agent to be approached at the realization s. If B_{σ_n} is coherent then σ_n surely chooses not to compute in $\mathcal{B}$. But this is not important as G is computed anyway.

Consider the other case, namely that B_{σ_n} is incoherent. B_{σ_n} can be partitioned into incoherent subsets, $A_{\sigma_n}^1, A_{\sigma_n}^2, \ldots, A_{\sigma_n}^t$, which are all in the image of the function A_{σ_n}, in $\mathcal{A}$.

As $\mathcal{A}$ is assumed appropriate the action of agent σ_n, in reply to any such subset, is to compute. So we deduce that $1 - c_{\sigma_n} \geqslant \operatorname{Prob}(Z_{\sigma_n} \mid A_{\sigma_n}^r) \cdot q + P(Z_{\sigma_n}^c \mid A_{\sigma_n}^r) \cdot 1$ for all indices r . By multiplying this formula by $\operatorname{Prob}(A_{\sigma_n}^r \mid B_{\sigma_n})$ and summing over r we conclude that $1 - c_{\sigma_n} \geqslant \operatorname{Prob}(Z_{\sigma_n} \mid B_{\sigma_n}) \cdot q + P(Z_{\sigma_n}^c \mid B_{\sigma_n}) \cdot 1$ which implies that it is optimal for σ_n to reply truthfully in $\mathcal{B}$.

Now we can use similar arguments for agent σ_{n-1} and so on. □

We now can phrase the main result for the general case:

Theorem 2. *Assume G is anonymous. Then there exists an appropriate mechanism at q if and only if there exists a truth-revealing Yes/No mechanism that is appropriate at q.*

Proof. Follows directly from Corollary 1 and Lemmas 1–3. □

The following example illustrates the power of Theorem 2.

Example 8. Consider a three agents problem and let us compute the number of mechanisms available:

- There are 3 options to choose the first agent, and for each of his two possible replies, one of the remaining two agents is chosen, yielding 4 options. As there is only a single option for the third agent, we conclude that there are $3 \cdot 2 \cdot 2 = 12$ options in total.
- Each agent, once approached can be provided with any subset of histories. As there are 27 possible histories in total, we conclude that the number of mechanisms is $12 \cdot 2^{27}$.

On the other hand the only degree of freedom, when considering truth-telling Yes/No mechanisms, is the order at which agents are approached. This implies that there are only 12 such possibilities.

3.1. Restricted case—fixed order mechanisms

The next result points out an optimal mechanism when the discussion is restricted to fixed order mechanisms.

Lemma 4. *Assume G is anonymous and $\mathcal{A}$ is a fixed order truth revealing Yes/No mechanism that is appropriate at q. Then the natural order Yes/No mechanism is also appropriate at q.*

Proof. Assume $\mathcal{A}$ is a fixed order Yes/No mechanism that is appropriate for G at q, but it is not the natural order Yes/No mechanism. Let k be the smallest index for which $k = \sigma_i > i$. Denote $m = \sigma^{-1}(i)$.

Denote by $\mathit{Permute}_{i \leftrightarrow k} : \Omega \to \Omega$ the permutation function that switches between the i and k entries.

Consider a new mechanism, $\mathcal{B} = (\tau, B)$, defined as follows:

- $\tau_j = \sigma_j$ for all $j \neq i, m$, $\tau_i = i$ and $\tau_m = k$.
- $B_j(h) = A_j(\mathit{Permute}_{i \leftrightarrow k}(h))$ for all $h \in H$.

We claim that the following strategy tuple, $\{X'_j\}_{j=1}^n$, is a computing equilibrium for $\mathcal{B}$. For all $\widetilde{H} \subset H$:

- For all agents $j \neq i, k$, set $X'_j(\widetilde{H}) = X_j(\mathit{Permute}_{i \leftrightarrow k}(\widetilde{H}))$
- $X'_i(\widetilde{H}) = X_k(\mathit{Permute}_{i \leftrightarrow k}(\widetilde{H}))$
- $X'_k(\widetilde{H}) = X_i(\mathit{Permute}_{i \leftrightarrow k}(\widetilde{H}))$

In words, all agents, but i and k, permute the information they receive and then follow the computing equilibrium strategy of $\mathcal{A}$. Additionally, agents i and k change roles. Once we show that $\{X'_j\}_{j=1}^n$ is an equilibrium it is immediate to verify it is actually a computing equilibrium.

To prove our claim we introduce additional notation as follows. Let $NO_j^{\mathcal{A}}$ (and $NO_j^{\mathcal{B}}$) be the incoherent set provided to the jth agent in $\mathcal{A}$ (respectively, $\mathcal{B}$) and $YES_j^{\mathcal{A}}$ (respectively, $YES_j^{\mathcal{B}}$) be its complement. Note that for each player σ_j, $Z_{\sigma_j} \subset NO_{j+j'}^{\mathcal{A}} \subset NO_j^{\mathcal{A}}$ for all $1 \leqslant j' \leqslant n - j$ (similarly for all τ_j), $Z_{\sigma_j} \subset NO_{j+j'}^{\mathcal{B}} \subset NO_j^{\mathcal{B}}$.

For all agents $j \neq i, k$ our claim is straightforward. What about agents i and k? Fix a specific realization of secrets and consider two cases as follows:

Case 1: Assume that for this realization agent i receives the information $YES_i^{\mathcal{B}}$. This means that k must receive the information $YES_m^{\mathcal{B}}$ in $\mathcal{B}$. Note that in the mechanism $\mathcal{A}$ agent k received the information $YES_i^{\mathcal{A}}$ (which equals $YES_i^{\mathcal{B}}$), and so his best reply is to guess, as dictated by X'_i. Furthermore, agent i must have received the information $YES_m^{\mathcal{A}}$, to which it replied by guessing. Therefore, X'_k dictates to guess, which is indeed agent k's best reply.

Case 2: Assume that for this realization agent i receives the information $NO_i^{\mathcal{B}}$. Therefore in $\mathcal{A}$, agent k would receive the information $NO_i^{\mathcal{A}} = NO_i^{\mathcal{B}}$ and would reply by computing. So his expected payoff from computing (recall it is a computing equilibrium) is $1 - c_k$, which is greater or equal than the expected payoff from guessing. We conclude that $1 - c_k \geqslant \text{Prob}(Z_k \mid NO_i^{\mathcal{A}}) \cdot q + \text{Prob}(Z_k^c \mid NO_i^{\mathcal{A}}) \cdot 1$. In $\mathcal{B}$, i faces the same distribution that k faced in $\mathcal{A}$ and so guessing would result in $\text{Prob}(Z_i \mid NO_i^{\mathcal{B}}) \cdot q + \text{Prob}(Z_i^c \mid NO_i^{\mathcal{B}}) \cdot 1$ as well. As $1 - c_i \geqslant 1 - c_k$ we learn that i's best reply, in $\mathcal{B}$, is to compute, which is exactly what is dictated by X_i'.

As for agent k, it may receive either the information $YES_m^{\mathcal{B}}$ or $NO_m^{\mathcal{B}}$. Assume it is the former; then on the one hand the best reply is to guess and on the other hand $X_k'(YES_m^{\mathcal{B}}) = X_i(Permute_{i \leftrightarrow k}(YES_m^{\mathcal{B}})) = X_i(YES_m^{\mathcal{A}})$, which is to guess, as well.

If agent k were to receive the information $NO_m^{\mathcal{B}}$ then X_k' dictates to compute. Lets verify this is indeed the best reply. By guessing k expects to receive $\text{Prob}(Z_k \mid NO_m^{\mathcal{B}}) \cdot q + \text{Prob}(Z_k^c \mid NO_m^{\mathcal{B}}) \cdot 1$. As $Z_k \subset NO_m^{\mathcal{B}} \subset NO_i^{\mathcal{B}}$ we know that guessing will generate an expected utility that is less or equal $\text{Prob}(Z_k \mid NO_i^{\mathcal{B}}) \cdot q + \text{Prob}(Z_k^c \mid NO_i^{\mathcal{B}}) \cdot 1$, which is equal $\text{Prob}(Z_k \mid NO_i^{\mathcal{A}}) \cdot q + \text{Prob}(Z_k^c \mid NO_i^{\mathcal{A}})$. The last term is known to be less or equal than $1 - c_k$, as we know that k computes when receiving the information $NO_i^{\mathcal{A}}$ in $\mathcal{A}$. We conclude that the payoff from guessing is less or equal that of computing and so reach the desired conclusion that computing is k's best reply.

Finally, note that if $\mathcal{B}$ does not have the natural order we can go on and switch one more pair of indices and introduce $\mathcal{C}$, and so on and so forth. We will eventually end up with the natural order Yes/No mechanism. □

A corollary of the last lemma is:

Theorem 3. *Assume G is anonymous, then there exists a fixed order appropriate mechanism at q, if and only if the natural order Yes/No mechanism is appropriate at q.*

Proof. Follows directly from Corollary 1 and Lemmas 1–4. □

Note that the number of fixed order, truth-telling, Yes/No mechanisms, for N agents is $N!$, whereas Theorem 3 suggest that only one mechanism should be studied when looking for an appropriate mechanism.

The last result suggests that perhaps a similar observation may hold for the general set of mechanisms. More formally:

Hypothesis. *If G is anonymous and there exists a mechanism that is appropriate at q then the Yes/No mechanism with natural order is also appropriate.*

Unfortunately, this is not the case as the following example demonstrates:

Example 9. Let G be the consensus function over 3 agents. Set $q = 0.5$ and $c_1 = 0, c_2 = 0.26, c_3 = 0.3$.

It is straightforward to verify that no fixed order Yes/No mechanism is appropriate, whereas the Yes/No mechanism where $\sigma_1 = 1$, $\sigma_2(0) = 2$ and $\sigma_2(1) = 3$ is appropriate.

To verify this note that player 1 computes as it has no cost for computing. As for player 3, note that if he gets *NO* signal from the mechanism it may be the case that $s_1 = 1$, with probability of $2/3$, or alternatively $s_1 = 0$ and $s_2 = 0$, with probability $1/3$. In the former case the probability of being pivotal is $1/2$ whereas in the second case it is 1. Thus, the expected payoff of guessing is $2/3$, which is less than $1 - 0.3$, the payment from computing. Similar considerations yield that player 2 computes.

Consider the functions AND and OR (the AND function equals 1 if and only if all secrets are 1, and the OR function equals 1 if and only if at least one secret is 1). Note that there exists a unique realization for which the mechanism will inform agents of the set NO_j, for all j. Therefore, a Yes/No mechanism necessarily has a fixed order (one can change the order of agents once the function can be computed to ensure this). Theorem 3 now implies that if there exists an appropriate mechanism for the AND (OR) function then the fixed order mechanism that approaches the agents according to the natural order is also appropriate.

3.2. Restricted case—equal cost structure

Our next result discusses the case where all costs are equal. We show that the simultaneous mechanism is equivalent to the Yes/No mechanism with a natural order.

Theorem 4. *If G is anonymous and costs are all equal then there exists an appropriate mechanism at q, if and only if the simultaneous mechanism is appropriate at q.*

Proof. One direction is obvious. Namely if the simultaneous mechanism is appropriate at q then there definitely exists an appropriate mechanism at q.

As for the other direction, let $\mathcal{A}$ be an arbitrary mechanism and assume it is appropriate for some probability q and c_j's. By our previous results we can assume, w.l.o.g., that $\mathcal{A}$ is the Yes/No mechanism.

Consider the first agent, σ_1. If he guesses, then by lemma 1 all subsequent agents will guess as well (and the function can already be computed) and therefore the function G must be constant, in which case the theorem's statement is obvious. Assume σ_1 actually computes his secret. Then we know that $1 - c_{\sigma_1} \geqslant \text{Prob}(Z_{\sigma_1} \mid A_1) \cdot q + P(Z^c_{\sigma_1} \mid A_1) \cdot 1$. However, as $A_1 = H$, $1 - c_j = 1 - c_{\sigma_1}$ and G is anonymous, we conclude that $1 - c_j \geqslant \text{Prob}(Z_{\sigma_1}) \cdot q + P(Z^c_{\sigma_1}) \cdot 1 = \text{Prob}(Z_j) \cdot q + P(Z^c_j) \cdot 1$.

We are done by noting that the condition for the simultaneous mechanism to be appropriate for q is exactly $1 - c_j \geqslant \text{Prob}(Z_j) \cdot q + P(Z^c_j) \cdot 1$. □

3.3. Restricted case—communication via broadcast channels

The model of the communication structure that is implicit in the definition of sequential mechanisms is that of 'private communication.' The mechanism is able to communicate with each agent over a private channel, without another agent eavesdropping. Consequently, the mechanism is able to pass on any information it chooses, when moving from one agent to another. It may reveal the exact nature of the previous replies or it may reveal partial information, or none at all.

Consider a communication structure consisting, solely, of a broadcast channel, one that allows everyone to communicate with everyone while all others may listen in. In this case agents will know who was approached and what was his reply. Therefore, mechanisms in such a communication environment, are necessarily 'fully revealing.' The flexibility of fully revealing mechanisms is limited to the choice of the order at which to approach agents.

We introduce the HCF (High Cost First) algorithm for (dynamically) ordering the agents:

- Step One—For each possible prefix compute the probability of being pivotal. Note that if G is anonymous then this is easy.
- Step Two—Consider the following recursive structure. For any given set of agents and costs choose the agent to move first as follows—Consider all agents with a cost low enough to justify computing (namely all j such that $1 - c_j \geqslant \text{Prob}(Z_j) \cdot q + P(Z_j^c) \cdot 1$) and approach the agent with the highest cost among these. Notice that computing the threshold cost is polynomial in the number of agents (recall the function id anonymous).
- Use this procedure to allocate the first agent (σ_1). Depending on the reply of σ_1 you end up with one of two trees. Apply the same procedure again to each tree, and so on and so forth.

It turns out that in such cases one can efficiently derive the appropriate mechanism, assuming such a mechanism exists. Notice that when we consider broadcast communication then the major issue in the definition of the mechanism is the order function. We can show:

Theorem 5. *Let G be an anonymous function. Assume it is common knowledge that there exists a fully revealing mechanism for G, then the fully revealing mechanism induced by the HCF algorithm has a computing equilibrium.*

Proof. We use an inductive argument, on the total number of agents. For $n = 1$ this is straightforward. Assume the algorithm holds for all problems with $n = N$ agents and consider a problem with $n = N + 1$ agents.

It is common knowledge that the original problem has an appropriate mechanism. Assume agent k is the first agent in that appropriate mechanism. We now know two things. First, that $1 - c_k \geqslant \text{Prob}(Z_k) \cdot q + P(Z_k^c) \cdot 1$, and second, that in the two problems induced, following k's reply, there are $n = N$ agents (all $N + 1$ agents, but k) and it is common knowledge that an appropriate mechanism exists.

Consider agent σ_1 that was chosen by our algorithm. By definition $c_{\sigma_1} \geqslant c_k$ (note that due to anonymity $\text{Prob}(Z_{\sigma_1}) \cdot q + \text{Prob}(Z_{\sigma_1}^c) \cdot 1 = \text{Prob}(Z_k) \cdot q + \text{Prob}(Z_k^c) \cdot 1$). Hence, if agent σ_1 moves first the existence of an appropriate mechanism for the two problems it induces is common knowledge.

Now, by our induction hypothesis, the algorithm induces a fully revealing appropriate mechanism for each of the two games induced by σ_1's reply, and so a computing equilibrium exists. It is therefore a best reply for agent σ_1 to compute as well. □

Remark. Note that the algorithm suggested can be implemented on-line. In other words, computation may take place only along the realized path. This is particularly interesting from a complexity point of view, as the number of histories along a particular path is n, whereas the total number of histories in H is of the order of magnitude of 2^n (and a naive off-line algorithm will have to refer to each of them).

The above result hinges on the existence of an appropriate mechanism. Verifying the actual existence of such a mechanism is a non trivial task. Indeed, one intuitive way to go about this question is to apply the algorithm suggested in the proof of Theorem 5 and eventually check whether the resulting mechanism is appropriate. However, this will be an exponential procedure. The discussion of the complexity of algorithms for verifying the existence of appropriate mechanisms is a subject for further study and will be discussed in a pending paper.

One may note the contrast between Theorem 5, which suggests approaching agents with high costs earlier, and Theorem 3, which suggests approaching agents with low costs earlier. This contrast stems from the fact that these two cases are, in some sense, orthogonal. Whereas the fixed order case, studied in Theorem 3, eliminates the possibility to choose the order of agents on-line and leaves the information revelation as the only degree of freedom on-line, the broadcast case does the opposite—it eliminates the decision on what information to disclose and leaves the order as the only degree of freedom.

4. Related literature

This paper considers multi-party computation from a game-theoretic perspective. Its novelty stems from treating multi-party computation as a public good setting, where costly computation of private inputs leads to free-riding problem, and from the introduction and study of mechanisms (and in particular sequential mechanisms) for overcoming this problem. Our framework therefore bridges the gap between work in multi-party computation in computer science and the literature on mechanism design and free-riding in economics and game theory. We now discuss some of the related literature, and position our work with respect to it.

4.1. Secure multi-party computation

Multi-party computation is a central topic in computer science. Many efforts have been devoted to the understanding of this topic. Of particular interest is the subject of secure multi-party computation (see e.g. Yao, 1982; Goldreich et al., 1987; Ben-Or et al., 1988, and a survey by Goldreich, 1998). The basic idea behind that literature is that a group of agents, each of which observes a private secret (e.g., bit) wishes to compute a function that is defined on these secrets (e.g. the majority of the bits' values). The main objective is to devise protocols for collective computation of the desired function value without having any information revealed to the parties, beyond the function's value. In particular, an agent will not be able to learn anything that is not implied by his private secret and the value of the function. The formal definitions of this setting follow the idea of zero-knowledge interactive proof systems of Goldwasser et al. (1988), a central topic in cryptography. Secure multi-party computation can be viewed as a form of a game between honest agents and malicious agents (see Linial, 1992 for a discussion of a game-theoretic perspective on that issue).

4.2. Game-theoretic multi-party computation

Although game-theoretic in nature, the work on secure multi-party computation does not include standard game-theoretic analysis. A complementary perspective, which does adopt a game-theoretic approach to multi-party computation, has been recently introduced by Shoham and Tennenholtz (2002). In their setting, titled *non-cooperative computing*, an agent's utility is effected by two factors: a primary objective of computing the function, and a secondary objective of preventing others from computing it. They provide full characterization of the boolean functions that can be non-cooperatively computed under various assumptions on the economic setting (e.g. private values vs. correlated values) and the algorithmic setting (e.g. deterministic vs. probabilistic algorithms). Most recently, work that attempts to combine secure multi-party computation and non-cooperative computing has been introduced by McGrew et al. (2003).

4.3. Multi-party computation and free riding

The above settings are all non-cooperative in nature, and consider agents with a strong externality: preventing the others from gathering information about other agents' secrets or/and the value of the function itself. In these settings each agent has some private information, accessible only by him at no cost. In this paper we expose and explore another fundamental aspect of multi-party computation. The main distinction of our setting from previous work on multi-party computation stems from the fact that the agents do not know their secrets to start with, and each agent has to spend some effort in obtaining his secret. The result of the computation is a public good and consequently we run into a free-riding problem. An agent may be tempted not to compute his own private value if this might not be pivotal for the function computation. This central aspect, of potential free-riding in multi-party computation, which is orthogonal to previous work on multi-party computation, is the subject of this paper.

The study of free riding has a long tradition in Economics and Game Theory, in particular in the context of the Public Good problem (e.g., Chapter 13 in Mas-Colell et al., 1995). In many models it is shown that free riding is sufficiently destructive to prevent socially optimal outcome (e.g., Rob, 1989 and Mailath and Postlewaite, 1990). In this paper, we take a more constructive approach and seek mechanisms that overcome the free riding problem and result in the efficient outcome, which is the correct execution of the multi-party computation. However, the results reported in this paper are restricted to the class of anonymous computations, namely computations that do not take into account agents' names. As in many Public Good problems the central consideration for the agents is whether or not their own secret has an impact on the joint computation. In other words, agents calculate how 'pivotal' they are. Pivotalness considerations are prevalent in many game theoretic models (e.g., Fudenberg et al., 1998; Al-Najjar and Smorodinsky, 2000). Such considerations also appear in problems studied by compute scientists, such as variable influence (e.g. Ben-Or and Linial, 1985 and Kahn et al., 1988) and t-resilient functions (e.g. Chor et al., 1985).

4.4. Mechanism design: entry fees

The basic model studied is that of asymmetric information. However, agents, initially, do not know their types (and so we depart from the classical 'private types' setting) but can access it, at a cost. Therefore, learning one's own secret (a-k-a type) is a strategic choice. An approach that resembles the one taken in this paper has been recently suggested in a voting model by Smorodinsky (2003). More generally, our work has some (although mostly superficial) similarity with assumptions taken in work on economic mechanism design. For example, work in auction theory distinguishes between private values, where each agent knows his type (e.g. valuation for an auctioned good), common value, where there is a shared value for an auctioned good which is initially unknown and is revealed only after the auction is over, and various intermediate cases (see Klemperer, 1999 for a recent overview). In our case, each agent has his own private type (secret), but he does not know its value (instantiation of the corresponding random variable); however, by applying some computation effort he can surely learn his type. This decision about learning an agent's own type is a strategic decision of the agent. This should be distinguished from work on mechanism design with entry fee (see Levin and Smith, 1994, for example) where agents need to decide whether to spend some fixed amount of money/effort in attending an auction. Although this decision may well influence the number of participants in an auction, it does not refer to a strategic decision about whether to learn one's type, and does not refer to the fundamental free riding problem associated with it.

Another paper which refers to the problem of a collective action with preliminary sunk costs (similar to the notion of entry fees), is the model studied in Admati and Perry (1991). They model a public good problem where agents pay up-front and bear their costs before the decision on supply takes place. They show inefficiency when the agents are approached sequentially in order to contribute. The Admati–Perry model studies a specific institution and involves two players who alternate in their moves.

4.5. Mechanism design: sequential mechanisms

Notice that from a technical perspective, our results can be categorized as contributions to the theory of mechanism design. However, in the mechanism design literature most models look at mechanisms that approach all agents simultaneously. Sequential mechanisms discussed in that literature are typically multi-stage games where the designer/center does not access agents sequentially, but the agents themselves may choose actions sequentially. In this paper we consider mechanisms that approach one agent after the other. These type of mechanisms turn out to be central in overcoming the free-riding problem in multi-party computations. An example for such sequential actions by the agents is the model of Marx and Matthews (2000) where agents' voluntary contributions to a public project are made incrementally. At each stage agents have information about the aggregate contribution and the mechanism plays no role in disseminating information. Another example is the above mentioned paper by Admati and Perry (1991).

4.6. Mechanism design: preference elicitation

Recent work in computer science has been concerned with algorithms for preference elicitation (see e.g. Boutilier et al., 1997 and Cohen and Sandholm, 2001). Our work contributes also to the literature on techniques for preference and information elicitation. However, although some of that work adopts a game-theoretic perspective (see e.g. Shoham and Tennenholtz, 2001), it does not deal with the fact private information might be costly to acquire, and with the fundamental free riding problem this issue introduces.

4.7. The revelation principle

A closely related branch of the literature is that on the revelation principle in Bayesian games (e.g., Myerson, 1982 and Section 6 in Myerson, 1991, among many others). Consider a model of incomplete information where agents communicate their information to an external mediator, who may in return suggest back to them some course of action. According to the general version of the revelation principle one may restrict attention to truth-telling agents, who comply with the mediator's recommended action. This is, intuitively, quite similar to the results of Theorems 1 and 2. One may interpret the Yes/No mechanism as a recommendation by the mechanism to the agents on whether to learn or to guess, and agents do comply, when mechanisms are appropriate.

There are two major reasons why the current model and results do not build on this revelation principle: (i) First, in our model, agents' information acquisition is a strategic decision and is, furthermore, costly to the agents. In Bayesian games, however, agents information is given to them (at no cost) and there is no acquisition stage (or, more accurately, the acquisition stage is not part of an agent's strategy). Note that agents decide on whether to acquire the information only after they get a recommendation from the mechanism. (ii) An important component in our model is the possibility to approach agents sequentially. The information revealed by agents approached early on is critical in determining the "recommendation" made to later agents. Example 2 demonstrates that a situation where a mechanism that simultaneously approaches all agents cannot simulate a sequential mechanism.

4.8. Collective decision making and the Condorcet Jury theorems

Collective decision making under uncertainty has been studied by various authors (e.g., Young, 1988; Austen-Smith and Banks, 1996 and Feddersen and Pesendorfer (1997, 1998)). In this setting committee members need to jointly decide which state of the world is correct. A leading example is that of a jury that is appointed to decide whether the defendant is guilty or innocent. Each committee member receives some noisy signal about the state and submits a vote accordingly. The goal of this literature is to find optimal voting rules, that minimize the probability of mistake. Often, the domain of search is the set of all majority mechanisms. Of particular interest is the comparison between simple majority and unanimity. Whereas collective decision making is at the heart of our model as well, we look at various functions and not only majority rules, and, furthermore, in our model information is costly.

Recent models look at situations where agents can acquire their information at a cost. Persico (2004), does so in a simultaneous decision making framework, Gerardi and Yariv (2004) extend Persico's model and allow for communication among agents. Finally, Gershkov and Szentes (2004) consider a sequential approach, similar to ours, and study efficient mechanisms, where efficiency takes into account the probability of correct decision making as well as the costs for acquiring information. In the work of Gershkov and Szentes, as well as in all previous papers, the mechanism is a mean to improve the probability of correct decision making. In our setting the goal is actually to compute some exogenously given function, and hence the main conceptual difference. Also, our approach allows for an arbitrary anonymous function and does not focus on majority rules.

An earlier model that studies sequential mechanisms in this context is that of Dekel and Dekel and Piccione (2000), who look at a sequential model of decision making but do not consider costly information acquisition. In fact, in the absence of this component, Dekel and Piccione show that sequential mechanisms provide nothing beyond what simultaneous mechanisms do.

5. Future research

This paper looks at the multi-party computation problem, from the perspective of mechanism design and individual incentives. It provides a general model and introduces 'sequential mechanisms.' This setup provides ground for a variety of interesting questions to pursue.

- *Natural extensions*: There are various ways to extend the model and questions discussed in this paper.
  - *Model of randomness*: In the model introduced here, agents' secrets are independent and identically distributed. Obviously this is quite restrictive. One can look at the general case, or perhaps at additional restricted cases, such as conditionally independent secrets, prevalent in the voting literature.
  - *Non-anonymous functions*: The study of anonymous functions, although well motivated, does exclude interesting functions such as electoral voting or weighted majority.
  - *Communication structure*: The generic model introduced in Section 2 assumes a private communication channel between the mechanism and the agents. However, in Section 3.3 we look at the implications of a more restricted model, that of a broadcast. In such a communications setup all messages are heard by everyone. This in turn implied that the mechanism must be fully revealing. Other natural restrictions can be studied. For example, assuming all down-links (i.e., from the mechanism/center to the agents) are public (broadcast), but up-links are private. Al-

ternatively, the mechanism may have access to some agents only via other agents (consider a model of a circular network for example).

- *Incentive structure/Externalities*: In this paper we look at a primitive incentive structure. Each agent may choose to pay or not for private information and may receive, or not, information on the value of G. No externalities, for example, are involved. However, agents' considerations may often depend on what opponents get. For example, whether or not they learned the agent's secret, or whether or not they learned the value of G (as in the paper of Shoham and Tennenholtz, 2002).
- *Existence of an appropriate mechanism*: This paper focuses on the structure of appropriate mechanisms. However, such mechanisms may fail to exist in some cases (e.g., apply the conclusion of Theorem 4 to Example 1). The existence of appropriate mechanisms, which is the subject matter of a pending paper, is a topic for further research. In particular, providing efficient algorithms or natural criteria to decide about the existence is an important topic. The characterization results of appropriate mechanisms, reported in this paper, provide an excellent starting point to study the existence question.
- *Probabilistic mechanisms*: In our setup mechanisms do not use lotteries to determine the order of agents or the information provided to them. All randomness follows from the secrets and agents' strategies. In fact, as demonstrated by the next example, probabilistic mechanisms are more powerful in the sense that there are cases where probabilistic mechanisms may be appropriate even when no (deterministic) mechanism is appropriate.

Example 10. Consider Example 9 without the agent with the zero cost. The Yes/No mechanism that uses a fair coin to determine the first agent is appropriate but any mechanism that chooses the first agent deterministically is not.

Acknowledgments

We thank the referees and an associate editor for their careful reading and helpful comments. Both authors acknowledge the support of the Bernard M. Gordon Center for Systems Engineering at the Technion. R. Smorodinsky acknowledges the support of the Japan Technion Society Research Fund.

References

Admati, A.R., Perry, M., 1991. Joint projects without commitment. Rev. Econ. Stud. 58, 259–276.

Austen-Smith, D., Banks, J.S., 1996. Information aggregation, rationality, and the Condorcet Jury theorem. Amer. Polit. Sci. Rev. 90, 34–45.

Al-Najjar, N.I., Smorodinsky, R., 2000. Pivotal players and the characterization of influence. J. Econ. Theory 92, 318–342.

Ben-Or, M., Linial, N., 1985. Collective coin flipping, robust voting schemes and minima of Banzhaf values. In: Proceedings of FOCS 1985, pp. 408–416.

Ben-Or, M., Goldwasser, S., Wigderson, A., 1988. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings 20th Annual ACM Symposium on Theory of Computing (STOC 88), pp. 1–10.

Boutilier, C., Brafman, R.I., Geib, C., Poole, D., 1997. A constraint-based approach to preference elicitation and decision making. In: AAAI Spring Symposium on Qualitative Decision Theory, March 1997.

Chor, B., Friedmann, J., Goldreich, O., Hastad, J., Rudich, S., Smolansky, R., 1985. The bit extraction problem or t-resilient functions. In: Proceedings of the 26th FOCS, pp. 396–407.

Cohen, W., Sandholm, T., 2001. Minimal preference elicitation in combinatorial auctions. In: International Joint Conference on Artificial Intelligence (IJCAI), Workshop on Economic Agents, Models, and Mechanisms, Seattle, WA, August 6th.

Dekel, E., Piccione, M., 2000. Sequential voting procedures in symmetric binary elections. J. Polit. Economy 108, 34–56.

Feddersen, T., Pesendorfer, W., 1997. Voting behavior and information aggregation in elections with private information. Econometrica 65, 1029–1058.

Feddersen, T., Pesendorfer, W., 1998. Convicting the innocent: The inferiority of unanimous jury verdicts under strategic voting. Amer. Polit. Sci. Rev. 92, 23–35.

Fudenberg, D., Levine, D., Pesendonfer, W., 1998. When are non-anonymous players negligible? J. Econ. Theory 79, 46–71.

Gerardi, D., Yariv, L., 2004. Committee design in the presence of communication. Working paper. UCLA.

Gershkov, A., Szentes, B., 2004. Optimal voting scheme with costly information acquisition. Working paper. Hebrew University.

Goldreich, O., 1998. Secure multi-computation. Working paper. Weizmann Institute.

Goldreich, O., Micali, S., Wigderson, A., 1987. How to play any mental game. In: Proceedings 19th Annual ACM Symposium on Theory of Computing (STOC 87), pp. 218–229.

Goldwasser, S., Micali, S., Rackoff, C., 1988. The knowledge complexity of interactive proof systems. SIAM J. Comp. (18), 186–208.

Kahn, J., Kalai, G., Linial, N., 1988. The influence of variables on boolean functions. In: Proceedings of FOCS 1988, pp. 68–80.

Klemperer, P., 1999. Auction theory: A guide to the literature. J. Econ. Surveys 13 (3), 227–286.

Levin, D., Smith, J.L., 1994. Equilibrium in auctions with entry. Amer. Econ. Rev. 84 (3), 585–599.

Linial, N., 1992. Games computers play: Game-theoretic aspects of computing. Technical report 92-5. Hebrew University.

Mailath, G.J., Postlewaite, A., 1990. Asymmetric information bargaining problems with many agents. Rev. Econ. Stud. 57, 351–367.

Marx, L.M., Matthews, S.A., 2000. Dynamic voluntary contribution to a public project. Rev. Econ. Stud. 67, 327–358.

Mas-Colell, A., Whinston, M.D., Green, J.R., 1995. Microeconomic Theory. Oxford Univ. Press, Oxford, New York (Chapter 11).

McGrew, R., Porter, R., Shoham, Y., 2003. Towards a general theory of non-cooperative computation. In: Proceedings of the 9th Conference of Theoretical Aspects of Rationality and Knowledge (TARK 2003), pp. 59–71.

Myerson, R., 1982. Optimal coordination mechanisms in generalized principal agent problems. J. Math. Econ. 10, 67–81.

Myerson, R., 1991. Game Theory—Analysis of Conflict. Harvard Univ. Press.

Persico, N., 2004. Committee design with endogenous information. Rev. Econ. Stud. 71, 165–192.

Rob, R., 1989. Pollution claim settlements under private information. J. Econ. Theory 47, 307–333.

Shoham, Y., Tennenholtz, M., 2002. Non-cooperative computation: Boolean functions with correctness and exclusivity. Unpublished manuscript. Stanford & Technion.

Shoham, Y., Tennenholtz, M., 2001. On rational computability and communication complexity. Games Econ. Behav. 35, 197–211.

Smorodinsky, R., 2003. A simple model of oligarchy. Unpublished manuscript. Technion.

Yao, A.C., 1982. Protocols for secure communication. In: Proceedings 27th IEEE Symposium on the Foundations of Computer Science (FOCS 82), pp. 16–164.

Young, P., 1988. Condorcet's theory of voting. Amer. Polit. Sci. Rev. 82, 1231–1244.