Sequential-Simultaneous Information Elicitation in Multi-Agent Systems*

Gal Bahar and Moshe Tennenholtz
Faculty of Industrial Engineering and Management
Technion–Israel Institute of Technology
Haifa 32000, Israel

Abstract

We introduce a general setting for information elicitation in multi-agent systems, where agents may be approached both sequentially and simultaneously in order to compute a function that depends on their private secrets. We consider oblivious mechanisms for sequential-simultaneous information elicitation. In such mechanisms the ordering of agents to be approached is fixed in advance. Surprisingly, we show that these mechanisms, which are easy to represent and implement are sufficient for very general settings, such as for the classical uniform model, where agents' secret bits are uniformly distributed, and for the computation of the majority function and other classical threshold functions. Moreover, we provide efficient algorithms for the verification of the existence of the desired elicitation mechanisms, and for synthesizing such mechanisms.

1 Introduction

Information elicitation in multi-agent systems deals with the aggregation of information from agents in order to compute a desired function [Conen and Sandholm, 2001; Sunderam and Parkes, 2003; Parkes, 1999; Boutilier *et al.*, 2003; Smorodinsky and Tennenholtz, 2004; Shoham and Tennenholtz, 2002; McGrew *et al.*, 2003; Halpern and Teague, 2004]. In the general setting we consider in this paper there are n agents (players) who hold some private information (secrets). The agents would like to compute, jointly, a function which input is the vector of n secrets.[1] In addition, there exist a distinguished reliable agent (aka the center), who may be used for extracting information from the agents. The center's aim is to devise a mechanism that will yield to the desired computation. In this paper we consider a strategic setting. We assume that the agents are selfish and are driven by utility maximization considerations. On one hand, they would like to receive the value of the multi-party computation, but on the other hand they may not want to contribute to this computation, but rather free-ride on other agents' efforts. Hence, our setting fits into the context of mechanism design (see [Mas-Colell *et al.*, 1995], Vol. 23), a subject of great interest to the synergy between AI and game theory (see [Monderer *et al.*, 2000]).

The particular (although general) setting we study is the following. Each agent has a secret which is accessible only to him. However, access to that secret is costly and agents may choose to access it or not. Thus 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.

As in previous work in this regard (e.g. [Smorodinsky and Tennenholtz, 2004]) the class of functions we consider 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 the 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, average, variance, order statistic, percentile and more.

Previous work has shown the power of sequential information elicitation. By approaching the agents sequentially, we can in some cases incent agents to access their secrets, where they would not do so if approached simultaneously. The basic intuition behind this approach is that some agents may be willing to access their secrets since this is not too costly for them, while others will not agree to access their secrets unless they are pivotal (given that their costs for accessing their secrets is high). Sequential mechanisms take advantage of this structure, by carefully selecting the identity of the agent to be approached at each point, given the available information. In this paper we consider a more general approach: agents can be approached both simultaneously and sequentially. That is, we consider sequential mechanisms where at each point a *set* of agents is approached. The decision on the set of agents to be approached is based on the available information. As it can be shown (and is illustrated in Section 2) this kind of mechanisms, which we refer to as *sequential-simultaneous* mechanisms, allows more effective information

*Supported in part by grant number 53/03-10.5 from the Israel Science Foundations

[1] This setting is fundamental not only to AI, but to computer science in general; see the discussion in [Linial, 1994].

elicitation than either sequential or simultaneous information elicitation. The major aim of this paper is to identify conditions for the existence of successful sequential-simultaneous elicitation mechanisms, and devising algorithms which output the desired mechanism when exist.

As the reader would notice, our objective is quite ambitious. In particular, a sequential-simultaneous mechanism may be a complex decision tree, which maps any possible history observed to the set of agents to be approached given that history. This gives rise to the idea of considering a particular kind of such mechanisms, which we refer to as *Oblivious Mechanisms*. In an oblivious mechanism the ordering of the agents is fixed in advance and does not depend on previous history. Naturally, if successful elicitation can be established by such a mechanism, then it is also easy to represent and implement. This leads to two types of fundamental questions:

1. Can one provide an efficient algorithm for checking whether, given an information elicitation setting, a desired oblivious sequential-simultaneous elicitation mechanism exist? If one exists, can we provide an efficient procedure which will output the desired mechanism?
2. Are there natural and general settings where the restriction to oblivious mechanisms does not prevent us from the desired elicitation?

The surprising message of this paper is that we can provide highly positive and encouraging answers to both questions! We first prove that if there is an appropriate oblivious mechanism then there is also one which uses a semi-natural ordering, i.e. the mechanism will never approach an agent with cost c before it approaches an agent with cost $c' < c$ (although it may approach them simultaneously). Then we show an efficient algorithm for verifying whether a desired (oblivious) semi-natural ordering mechanism exists; if such mechanism exists, then the desired ordering is obtained as our algorithm's output. Then, we show that the restriction to (oblivious) semi-natural ordering mechanism does not restrict the power of elicitation in some of the most popular contexts. In particular, we show that this holds for the uniform setting (where each agent's secret bit is taken to be 0 or 1 with probability 0.5 each), as well as for the classical majority function and for more general threshold functions. Together, this provides a most powerful and general approach to information elicitation in multi-agent systems, backed up with efficient and effective algorithms.

In Section 2 we introduce the general model, and general sequential-simultaneous mechanisms. We also briefly illustrate the power that these mechanisms buy us when comparing to previous studies where purely sequential or purely simultaneous mechanisms have been considered. In Section 3 we introduce oblivious mechanisms. We then show an efficient algorithm for verifying the existence of an appropriate oblivious elicitation mechanism, and show that a semi-natural ordering mechanism can be computed as the desired result, if such oblivious mechanism exists. We also remark (by means of example) that there are cases where the restriction to oblivious mechanisms may prevent us from obtaining the desired computation; this leads to the study of settings where the existence of a desired elicitation process implies that it can be done by a corresponding oblivious mechanism, which we can efficiently compute. The uniform model is discussed in Section 4, and the majority function, as well as extensions to other threshold functions are discussed in Section 5. In all cases efficient algorithms are provided.

The proofs of many of our results are non-trivial, and are based on a series of propositions that are omitted from this version of the paper due to lack of space.

2 The General Model

Let N={1,....,n}, $n \geq 2$, be a finite set of agents. Each agent j has a private secret $s_j \in \{0,1\}$ that he may compute (i.e. access in order to learn its value). Let $0.5 \leq q < 1$ be the prior probability of $s_j = 1$[2] and assume these events are independent. Agents may compute their own secrets. However, computation is costly and agent j pays $c_j \geq 0$ for computing s_j. Without loss of generality we shall assume $c_1 \leq c_2 \leq \ldots \leq c_n$ (in words, agents are ordered by their costs). We will also refer to the agents' costs, ordered from the lowest one to the highest one, as $c_{(0)}, c_{(1)}, \ldots, c_{(n-1)}$.

Agents are interested in computing some joint binary parameter (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 learning the value of G. We assume $v_j \geq c_j$, otherwise the agent face no dilemma.

It is also possible to 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 $\frac{c_j}{v_j}$.

A central designer, termed the *center*, 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.

One should note that the cost of each agent's computation, c_j, is lower than the gain from computing G, and therefore the socially optimal outcome is to compute. However, free riding of agents may undermine the ability to reach the social optimum.

2.1 Sequential-Simultaneous mechanisms

We now model mechanisms that approach players sequentially, with the possibility to approach simultaneously more than one player each time. We will assume that the communication between the center and the agents is fully revealed to all agents. This can be associated with having broadcast communication.

A sequential-simultaneous mechanism determines the set of agents to be approached at each stage, as a function of the information provided so far. Formally, let $H_i =$

[2]This is done wlog and for purposes of exposition. The case where $q < 0.5$ is treated similarly.

$\{(a_1,b_1),(a_2,b_2),\ldots,(a_i,b_i)|a_i \le n, \forall 1 \le k < i : a_k < a_{k+1}, \forall 1 \le k < i : b_k \le b_{k+1}, \forall 1 \le k < i : a_{k+1} - a_k \ge b_{k+1} - b_k\}$ where $v = (j,l)$ stands for the event where j agents have been approached and l 1's (and $j-l$ 0's) have been reported. H_i is the set of histories of length i, for $1 \le i \le n$. Let $H_0 = \Lambda$, where Λ is the empty (null) history. Let $H = \bigcup_{i=0}^{n} H_i$ be the set of all possible histories . A sequential-simultaneous mechanism is a triplet (u,g,f) where $u : H \to \{1,2,\ldots,n\}$ determines how many agents to approach simultaneously, $g : H \to 2^N$ determines the agents to be approached, and $f : H \to \{0,1,*\}$ is a function that expresses a decision about whether to halt and output either 0 or 1, or continue the elicitation process (denoted by *). Note that each agent has 6 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 strategy for player j, $x_j : H \to \Gamma$, assigns an action to each possible history. Let $h_{(k)}(x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n,S)$ be the history that agent k sees when approached, given that $(x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n)$ is the vector of strategies of the other agents and S is the vector of secrets. Let $x_k(h_{(k)})$ be the strategy of agent k according to the history $h_{(k)}$ he sees when approached. Let $x_k^1(h_{(k)}) = 1$ if and only if the action of agent k according to the history $h_{(k)}$ is to compute the secret, otherwise $x_k^1(h_{(k)}) = 0$, and let $x_k^0(h_{(k)}) = 1$ if and only if agent k reports "1", otherwise $x_k^0(h_{(k)}) = 0$. We get that the utility for agent k according to the strategies and the secrets is: $u_k(x_1,\ldots,x_n) = [Prob(G(x_1^0(h_{(1)}),\ldots,x_n^0(h_{(n)})) = G(S)] - [Prob(x_k^1(h_{(k)})) = 1] \cdot c_k$ [3]. An equilibrium for the mechanism $\mathcal{A}$, is a vector of n strategies, $(x_1,\ldots,x_n)$, such that each agent's strategy is the best response against the other agents' strategies. That is, for each agent k and for each strategy x'_k of agent k: $u_i(x_1,\ldots,x_k,\ldots,x_n) \ge u_i(x_1,\ldots,x'_k,\ldots,x_n)$. 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 access and reveal their secrets, 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.

Given a particular q and $c_1,\ldots,c_n$, a mechanism $\mathcal{A}$ is *appropriate* for G, 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. An algorithm is appropriate if it induces an appropriate mechanism.

Let G be an anonymous function. Consider the directed graph $G' = (V',E')$ where the set of nodes is $V' = \{(i,k) : i,k \in Z_+, 0 \le k \le i \le n-1\}$, and the set of edges is, $E' = \{((i,k),(i+1,k)) : 0 \le i \le n-2, k \le i\} \bigcup \{((i,k),(i+1,k+1)) : 0 \le i \le n-2, k \le i\}$.

[3] The probability distribution is induced by the distribution over S, and q.

Intuitively, this graph describes the possible states of information, when approaching agents while computing the value of an anonymous function, and the possible state transitions. Every state describes how many agents have been approached and how many 1's have been heard so far. Let $\bar{G} = (\bar{V},\bar{E})$ be the graph induced by reducing G' to include only nodes where the value of the function G can not be determined yet. When referring to a node v, we will abuse notation and refer also to v as an event. Namely, the event $v = (j,l)$ stands for the case that j agents have been approached and l 1's (and $j-l$ 0's) have been reported.

For every node v in $\bar{G}$ we define the following. Let $piv(v,G) = Prob(Z_i|v)$, where Z_i is the event that agent i is pivotal for the function G. Notice that since G is an anonymous function $Prob(Z_i|v)$ is the same for each agent i and therefore $piv(v,G)$ is uniquely defined. When G is clear from the context then denote it $piv(v)$. Note that if $v \notin \bar{V}$, i.e. it is already possible to compute the value of G, then $piv(v) = 0$. Let $Cmax(v) \equiv Max\{c \mid 1-c \ge piv(v)\cdot q + (1-piv(v))\cdot 1\}$. The left side of the above inequality is the utility for an agent with cost c from approaching his secret assuming all other agents approach their secrets. The right side is the utility for the agent when he doesn't approach the secret (assuming all other agents approach their secrets) and guess "1", which is the best choice for him in this case (since $q \ge 0.5$). This inequality implies that: $c \le 1-(1-piv(v)) - q\cdot piv(v)$. This implies that $c \le (1-q)\cdot piv(v)$. Hence we get: $Cmax(v) = (1-q)\cdot piv(v)$. Notice that $Cmax(v)$ is the maximal cost for which it is still rational to access the secret given the information v (assuming the others access their secrets).

The following example illustrates why is it useful to use sequential-simultaneous mechanisms. Consider the following setup. Let G be the majority function, $n = 3$, $q = 0.7$, and $c_{(0)} = 0.12$, $c_{(1)} = 0.12$, $c_{(2)} = 0.29$. We can show: given the above setup there exists a sequential-simultaneous mechanism which is appropriate. On the other hand, given the above setup there does not exist a sequential mechanism which is appropriate. It is also the case that approaching all agents simultaneously is not appropriate.

3 Oblivious mechanisms

An *oblivious mechanism* is a mechanism that partition the agents to groups and each time approach the agents from one group simultaneously without dependency on the history. Both the groups and the ordering between the groups are determined in advance. That is, the agents are partitioned into groups: $A_1, A_2, \ldots A_t$. such that: $\bigcup_{i=1}^{t} A_i = \{1,2,..,n\}$, $\forall\ 1 \le i \ne j \le t : A_i \bigcap A_j = \emptyset$ where the group A_1 of agents is approached first, and afterwards the group A_2 of agents is approached, and so on until the group A_t of agents is approached. However, if after a group of agents is approached and reveal their secrets, the event is such that it is already possible to determine the value of the function, then the mechanism stops.

Given a function G, agents' costs $c_{(0)} \le c_{(1)} \le \ldots \le c_{(n-1)}$, and q, a *semi-natural ordering mechanism* is an oblivious mechanism, where the agents are also sorted accord-

ing to their cost for accessing their secrets. That is, $A_1 = \{c_{(0)}, c_{(1)}, \ldots, c_{(|A_1|-1)}\}$,
$,A_2 = \{c_{(|A_1|)}, c_{(|A_1|+1)}, \ldots, c_{(|A_1|+|A_2|-1)}\}, \ldots,$
$\ldots,$
$A_{t-1} = \{c_{(|A_1|+\cdots+|A_{t-2}|)}, c_{(|A_1|+\cdots+|A_{t-2}|+1)}, \ldots, c_{(|A_1|+\cdots+|A_{t-2}|+|A_{t-1}|-1)}\}$,
$A_t = \{c_{(|A_1|+\cdots+|A_{t-1}|)}, c_{(|A_1|+\cdots+|A_{t-1}|+1)}, \ldots, c_{(n-1)}\}$

This section deals with the following: First we prove that whenever there exists an oblivious mechanism which is appropriate, there exists a semi-natural ordering mechanism which is appropriate. We then address the problem of existence: We provide an efficient algorithm, that given an anonymous function G, the c_i's, and q, will decide if there exist an oblivious mechanism which is appropriate. Moreover, we provide an efficient algorithm that induces an appropriate oblivious mechanism whenever such a mechanism exist.

Given a function G, agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q, consider the following algorithm.

The Oblivious Mechanism Algorithm: (OMA)

(A) Build the graph $G' = (V', E')$ of the function G

(B) Compute for every node $v \in V' : piv(v)$ and $Cmax(v)$.

(C) Induce the graph $\bar{G} = (\bar{V}, \bar{E})$ by reducing G' to include only nodes where the value of the function G can not be determined yet.

(D) $\forall\ 0 \leq k \leq n-1$: let $L_k = Min\{Cmax(k,j)|(k,j) \in \bar{V}\}$ i.e. L_k is the minimum $Cmax$ among all the nodes with the same depth k in $\bar{G}$.

(E) Let $i_1 = 0$.
$i_2 = Min\{j|i_1 < j \leq n-1, L_j > L_{i_1}\}$.
$i_3 = Min\{j|i_2 < j \leq n-1, L_j > L_{i_2}\}$.
...
$i_r = Min\{j|i_{r-1} < j \leq n-1, L_j > L_{i_{r-1}}\}$.
...
$i_k = Min\{j|i_{k-1} < j \leq n-1, L_j > L_{i_{k-1}}\}$ where i_k satisfies $\forall j :\ i_k < j \leq n-1 : L_j \leq L_{i_k}$

(F) Let $g_1 = i_2 - i_1, g_2 = i_3 - i_2, \ldots, g_{r-1} = i_r - i_{r-1}, \ldots, g_{k-1} = i_k - i_{k-1}, g_k = n - i_k$.

(G) Sort the agents according to their costs of getting to their secrets and get: $c_{(0)} \leq c_{(1)} \leq ... \leq c_{(n-1)}$

(H) Check if the following set of conditions, marked as *Cond1*, is satisfied: $c_{(0)}, c_{(1)}, ..., c_{(g_1-1)} \leq L_{i_1}$
$c_{(g_1)}, c_{(g_1+1)}, ..., c_{(g_1+g_2-1)} \leq L_{i_2}$
$c_{(g_1+g_2)}, c_{(g_1+g_2+1)}, ..., c_{(g_1+g_2+g_3-1)} \leq L_{i_3}$
$\cdots$
$c_{(g_1+g_2+\ldots+g_{k-2})}, \ldots, c_{(g_1+g_2+\cdots+g_{k-1}-1)} \leq L_{i_{k-1}}$
$c_{(g_1+g_2+\cdots+g_{k-2}+g_{k-1})}, \ldots, c_{(g_1+g_2+\cdots+g_{k-1}+g_k-1)} \leq L_{i_k}$

Consider the set of conditions *Cond1* in step (H) of the OMA above. We can now prove the following general results:

Theorem 3.1: Consider an anonymous function G, agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q. If there exists an oblivious mechanism which is appropriate then *Cond1* is satisfied.

Theorem 3.2: Consider an anonymous function G, agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q, if *Cond1* is satisfied then there exists a semi-natural ordering mechanism which is appropriate.

Note that given an anonymous function function G, agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q, if there exists a semi-natural ordering mechanism which is appropriate then there exists an oblivious mechanism which is appropriate. Therefore, combining the above theorems and observations we get:

Theorem 3.3: Consider an anonymous function G, agents costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q. The three following claims are equivalent: (1) There exists an oblivious mechanism which is appropriate. (2) *Cond1* is satisfied. (3) There exists a semi-natural ordering mechanism which is appropriate.

Now we can provide a complete efficient procedure that given a function G, agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q, will check if there exists an oblivious mechanism which is appropriate. Moreover, if there exist an oblivious mechanism which is appropriate, this procedure will induce an appropriate semi-natural ordering mechanism. In order to do so we repeat steps (A) - (G) of the OMA. Then, as in step (H) of of OMA we check whether *Cond1* is satisfied; if *Cond1* is not satisfied then the procedure will return "there isn't any oblivious algorithm which is appropriate" and finish; otherwise, if *Cond1* is satisfied then the mechanism will approach the agents sequentially, in the following order, where at each phase we approach simultaneously a set of agents as described below (the agents are referred to by their costs):

- $c_{(0)}, c_{(1)}, ..., c_{(g_1-1)}$.
- $c_{(g_1)}, c_{(g_1+1)}, \ldots, c_{(g_1+g_2-1)}$.
- $c_{(g_1+g_2)}, c_{(g_1+g_2+1)}, \ldots, c_{(g_1+g_2+g_3-1)}$.
- $\cdots$
- $c_{(g_1+\cdots+g_{k-2})}, c_{(g_1+\cdots+g_{k-2}+1)}, \ldots, c_{(g_1+\cdots+g_{k-2}+g_{k-1}-1)}$
- $c_{(g_1+\cdots+g_{k-1})}, c_{(g_1+\cdots+g_{k-1}+1)}, \ldots, c_{(g_1+\cdots+g_k-1)} = c_{(n-1)}$.

To sum up, given a function G, agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q, the OMA algorithm verifies the existence of an appropriate oblivious mechanism. Such a mechanism, if exists, is induced by the above procedure.

Oblivious mechanisms are natural and can be easily implemented. It can be shown (by means of an example; details omitted due to lack of space) that the existence of an appropriate mechanism does not necessary imply the existence of an oblivious mechanism which is appropriate. However, surprisingly, in the next sections we show that there are some important and central setups where oblivious mechanisms are sufficient for efficient and effective information elicitation. Given the convenient and tractable structure of these mechanisms, this provides powerful tools for information elicitation. Namely, this turned out to be the case for the classical uniform model discussed in Section 4, and for the majority function, and moreover in the context of threshold functions, as discussed in Section 5.

4 The Uniform Model

Consider the case where $q = 0.5$ and G is any anonymous function. The assumption that the distribution is uniform is most standard in many basic studies in this context. Consider the following algorithm:

The Uniform Semi-Natural Ordering Algorithm (USNO):

(A) Build the graph $G' = (V', E')$ of the function G.

(B) Compute $\forall\, v \in V' : piv(v)$ and $Cmax(v)$.

(C) Induce the graph $\bar{G} = (\bar{V}, \bar{E})$ by reducing G' to include only nodes where the value of the function G can not be determined.

(D) Denote as Υ the following path in $\bar{G}$: $(0,0) = v'_0 \to v'_1 \to v'_2 \to \ldots \to v'_j = (j,l) \to v'_{j+1} \to \ldots \to v'_{n-1}$ where $(0,0) = v'_0 \to v'_1 \to v'_2 \to \ldots \to v'_j = (j,l)$ satisfies that $v'_j = (j,l)$ has only one son in $\bar{G}$, and $\forall\ 0 \le i \le j-1 : piv(v'_i) \ge piv(v'_{i+1})$. Let $v'_j = (j,l) \to v'_{j+1} \to \ldots \to v'_{n-1}$ be the suffix from v'_j to the only leaf in $\bar{G}$ reachable from v'_j. The existence of such a path is induced by a series of propositions omitted from the paper due to lack of space.

(E) Along Υ, $\forall\ 0 \le i \le n-1$: We will mark $\Psi_i \equiv Cmax(v'_i)$

(F) Check which is the first node on Υ that satisfies $\Psi_t > \Psi_0$ [4]. That is, $t = Min\{k|\ 1 \le k \le n-1,\ \Psi_k > \Psi_0\}$. (the fact that $t > j$ is implied by the above mentioned propositions).

(G) Sort the agents according to their costs of getting to their secrets and get: $c_{(0)} \le c_{(1)} \le ... \le c_{(n-1)}$

(H) Check if the following is satisfied: $c_{(0)} \le \Psi_0$, $c_{(1)} \le \Psi_0, \ldots$, $c_{(t-1)} \le \Psi_0$, $c_{(t)} \le \Psi_t$, $c_{(t+1)} \le \Psi_{t+1}, \ldots$, $c_{(n-1)} \le \Psi_{n-1}$. If the above is not satisfied then finish and return "there is no appropriate mechanism for the problem", else continue to phase (I).

(I) First approach simultaneously the t agents with the lowest costs, i.e. $c_{(0)}, c_{(1)}, \ldots, c_{(t-1)}$. Afterwards approach each time one agent with the lowest cost among those left, i.e. first approach $c_{(t)}$,then approach $c_{(t+1)}$, etc, when $c_{(n-1)}$ is approached last.

We can now prove:

Theorem 4.1: Consider an anonymous function G, agents' costs $c_{(0)} \le c_{(1)} \le \ldots \le c_{(n-1)}$, and $q = 0.5$, and let $\Psi_i \equiv Cmax(v'_i)$ for every $0 \le i \le n-1$ along the path Υ. Let $t = min\{k|\ 1 \le k \le n-1,\ \Psi_k > \Psi_0\}$. Then there exists an appropriate algorithm, if and only if: $c_{(0)} \le \Psi_0$, $c_{(1)} \le \Psi_0, \ldots, c_{(t-1)} \le \Psi_0$, $c_{(t)} \le \Psi_t$, $c_{(t+1)} \le \Psi_{t+1}, \ldots, c_{(n-1)} \le \Psi_{n-1}$.

Theorem 4.2: Consider an anonymous function G, agents' costs $c_{(0)} \le c_{(1)} \le \ldots \le c_{(n-1)}$, and $q = 0.5$, if there exists an appropriate mechanism then the USNO algorithm induces an appropriate oblivious mechanism.

Corollary 4.1: Consider an anonymous function G, agents' costs $c_{(0)} \le c_{(1)} \le \ldots \le c_{(n-1)}$, and $q = 0.5$. The USNO handles the problem of existence, i.e the USNO will efficiently compute whether there exists any appropriate mechanism. If such a mechanism exists, the USNO will output an appropriate semi-natural ordering mechanism.

5 Computing the majority function

The majority function is central to voting theory, and as a result to the related multi-agent systems and distributed computing literature. Consider the case where n is odd, and $q > 0.5$ and the function we would like to compute is the majority function, [5] i.e. if $\lceil \frac{n-1}{2} \rceil + 1$ or more of the agents' secrets are 1's then the value of the function will be 1, else the value will be 0.

We will now present an efficient algorithm that induces an appropriate mechanism if such a mechanism exists. The output mechanism will be an oblivious, semi-natural ordering mechanism. Denote by Γ the following path in the graph associated with the majority function:
$v^*_0 = (0,0) \to (1,1) \to (2,2) \to \ldots \to (\frac{n-1}{2}, \frac{n-1}{2}) \to (\frac{n-1}{2}+1, \frac{n-1}{2}) \to (\frac{n-1}{2}+2, \frac{n-1}{2}) \to \ldots \to (n-1, \frac{n-1}{2}) = v^*_{n-1}$

The Semi-Natural Ordering Algorithm (SNO):

(A) Compute: $\forall\ 0 \le i \le n-1$: $piv(v^*_i)$ and $Cmax(v^*_i)$, along Γ. $\forall\ 0 \le i \le n-1$: We will mark $\Psi_i \equiv Cmax(v^*_i)$

(B) Check which is the first node on Γ that satisfies $\Psi_j > \Psi_0$. That is, $j = min\{k|\ 1 \le k \le n-1,\ \Psi_k > \Psi_0\}$.[6].

(C) Sort the agents according to their costs of getting to their secrets and get: $c_{(0)} \le c_{(1)} \le ... \le c_{(n-1)}$

(D) Check if the following is satisfied: $c_{(0)} \le \Psi_0$, $c_{(1)} \le \Psi_0, \ldots$, $c_{(j-1)} \le \Psi_0$, $c_{(j)} \le \Psi_j$, $c_{(j+1)} \le \Psi_{j+1}, \ldots$, $c_{(n-1)} \le \Psi_{n-1}$. If the above is not satisfied then finish and output "there is no appropriate mechanism for the problem", else continue to phase (E).

(E) First approach simultaneously the j agents with the lowest costs, i.e. $c_{(0)}, c_{(1)}, \ldots, c_{(j-1)}$. Afterwards approach each time one agent with the lowest cost among those left, i.e. first approach $c_{(j)}$,then approach $c_{(j+1)}$, etc, when $c_{(n-1)}$ is approached last.

We can now show:

Theorem 5.1: Consider the majority function G, agents' costs $c_{(0)} \le c_{(1)} \le \ldots \le c_{(n-1)}$, and q, and let $\Psi_i \equiv Cmax(v^*_i)$ for every $0 \le i \le n-1$ along the path Γ. Let $j = min\{k|\ 1 \le k \le n-1,\ \Psi_k > \Psi_0\}$. Then there exists an appropriate algorithm, if and only if: $c_{(0)} \le \Psi_0$, $c_{(1)} \le \Psi_0, \ldots, c_{(j-1)} \le \Psi_0$, $c_{(j)} \le \Psi_j$, $c_{(j+1)} \le \Psi_{j+1}, \ldots$, $c_{(n-1)} \le \Psi_{n-1}$.

Theorem 5.2: Consider the majority function G, agents' costs $c_{(0)} \le c_{(1)} \le \ldots \le c_{(n-1)}$, and q, if there exists an appropriate mechanism then the SNO algorithm induces an appropriate oblivious mechanism.

---

[4] It is guaranteed that there exists such a node if G is not the parity or the $\neg$ parity function. These functions are treated separately, leading to similar results.

[5] Our results in this section can be extended in a straightforward manner to the case where $q \le 0.5$ but not hold to the case where n is even; the case where n is even will be discussed later, when we extend the discussion to general threshold functions.

[6] As before, the fact that such a j exists is implied by a series of propositions characterizing the properties of Γ

Corollary 5.1: Consider the majority function G, agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$, and q, the SNO handles the problem of existence, i.e the SNO will efficiently compute whether there exists any appropriate mechanism. If such a mechanism exists, the SNO will output an appropriate semi-natural ordering mechanism.

The computation of majority when $q < 0.5$: Consider the case where n is odd, and the function that we would like to compute is the majority function where $q < 0.5$. In this case we can use similar techniques to the ones we used for the computing of majority where $q > 0.5$. The main difference is that instead of the path Γ we will consider the path Γ': $v_0^{'} = (0,0) \rightarrow (1,0) \rightarrow (2,0) \rightarrow \ldots \rightarrow (\frac{n-1}{2}, 0) \rightarrow (\frac{n-1}{2} + 1, 1) \rightarrow (\frac{n-1}{2} + 2, 2) \rightarrow \ldots \rightarrow (n-1, \frac{n-1}{2}) = v_{n-1}^{'}$. Note that the case where $q = 0.5$ is covered by the uniform model section.

5.1 Threshold functions

Given an anonymous function G. Let $G(T)$ be the value of the function G when there are T "1"'s and $n - T$ "0"'s. G is considered as a *threshold function* if there exists $1 \leq r \leq n$ such that: $G(0) = G(1) = ... = G(r-1) \neq G(r) = G(r+1) = ... = G(n)$, i.e. whenever there are less than r 1's among the agents then the function gets a particular value, and where there are r or more 1's the value of the function will be the other value. The majority function for example is a threshold function with $r = \lceil \frac{n+1}{2} \rceil$ where $G(0) = 0$.

We can now prove the following general theorems:

Theorem 5.3: Consider an anonymous threshold function G with threshold r where $\lceil \frac{n+1}{2} \rceil \leq r \leq n$, $q \leq 0.5$ and agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$. There exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

Theorem 5.4: Consider an anonymous threshold function G with threshold r where $\lceil \frac{n+1}{2} \rceil \leq r \leq n$, $q \geq \frac{r-1}{n-1}$ and agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$. There exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

Similarly to theorems 5.3 and 5.4, we can also show:

Theorem 5.5: Consider an anonymous threshold function G with threshold r where $1 \leq r \leq \lfloor \frac{n+1}{2} \rfloor$, and $q \leq \frac{n-r}{n-1}$ or $q \geq 0.5$ and agents' costs $c_{(0)} \leq c_{(1)} \leq \ldots \leq c_{(n-1)}$. Then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

Further discussion about majority: The majority function is a particular case of a threshold function, where $r = \lceil \frac{n+1}{2} \rceil$. However when n is odd, we get $r = \frac{n+1}{2}$. Theorem 5.3 implies that if $q \leq 0.5$ then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate. Theorem 5.4 implies that if $q \geq \frac{r-1}{n-1} = \frac{\frac{n+1}{2}-1}{n-1} = \frac{\frac{n-1}{2}}{n-1} = 0.5$ then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate. Hence when n is odd the above holds for every q (as we have already shown before).

Consider the case where G is the majority function and n is even. This implies that $r = \frac{n+2}{2}$. Theorem 5.3 implies that if $q \leq 0.5$ then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate. Theorem 5.4 implies that if $q \geq \frac{r-1}{n-1} = \frac{\frac{n+2}{2}-1}{n-1} = \frac{\frac{n}{2}}{n-1} = 0.5 \cdot \frac{n}{n-1}$ then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate. However, when n is large this implies that for almost every q there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

References

[Boutilier *et al.*, 2003] C. Boutilier, R. Das, J.O. Kephart, G. Tesauro, and W.E. Walsh. Cooperative negotiation in autonomic systems using incremental utility elicitation. In *Proceedings of UAI-03*, pages 89–97, 2003.

[Conen and Sandholm, 2001] W. Conen and T. Sandholm. Minimal preference elicitation in combinatorial auctions. In *IJCAI Workshop on Economic Agents, Models, and Mechanisms, Seattle, WA, August 6th.*, 2001.

[Halpern and Teague, 2004] J. Halpern and V. Teague. Rational secret sharing and multiparty computation. In *Proc. of STOC 2004*, 2004.

[Linial, 1994] N. Linial. Game-Theoretic Aspects of Computer Science. In R.J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications, Vol. II, Chapter 38*, pages 1340–1395. North-Holland, 1994.

[Mas-Colell *et al.*, 1995] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[McGrew *et al.*, 2003] R. McGrew, R. Porter, and Y. Shoham. Towards a general theory of non-cooperative computation. In *Proc. of the 9th conference of theoretical aspects of rationality and knowlede (TARK 2003)*, pages 59–71, 2003.

[Monderer *et al.*, 2000] D. Monderer, M. Tennenholtz, and H. Varian. Game theory and artificial intelligence. Special issue of Games and Economic behavior, 2000.

[Parkes, 1999] D. C. Parkes. Optimal auction design for agents with hard valuation problems. In *Proc. IJCAI-99 Workshop on Agent Mediated Electronic Commerce*, pages 206–219, 1999.

[Shoham and Tennenholtz, 2002] Y. Shoham and M. Tennenholtz. Non-cooeartive computation: Boolean functions with correctness and exclusivity. Technical Report, Stanford & Technion; To appear in Theoretical Computer Science, 2002.

[Smorodinsky and Tennenholtz, 2004] Rann Smorodinsky and Moshe Tennenholtz. Sequential information elicitation in multi-agent systems. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 528–535, 2004.

[Sunderam and Parkes, 2003] Aditya V Sunderam and David C. Parkes. Preference elicitation in proxied multiattribute auctions. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, pages 214–215, 2003.