Exponential Communication Inefficiency of Demand Queries*

Noam Nisan† Ilya Segal‡

May 10, 2005

Abstract

In the problem of finding an efficient allocation when agents' utilities are privately known, we examine the effect of restricting attention to mechanisms using "demand queries," which ask agents to report an optimal allocation given a price list. We construct a combinatorial allocation problem with m items and two agents whose valuations lie in a certain class, such that (i) efficiency can be obtained with a mechanism using $O(m)$ bits, but (ii) any demand-query mechanism guaranteeing a higher efficiency than giving all items to one agent uses a number of queries that is exponential in m. The same is proven for any demand-query mechanism achieving an improvement in *expected* efficiency, for a constructed joint probability distribution over agents' valuations from the class. These results cast doubt on the usefulness of such common combinatorial allocation mechanisms as "iterative auctions" and other "preference elicitation" mechanisms using demand queries, as well as "value queries" and "order queries" (which are easily replicated with demand queries in our setting).

1 Introduction

We consider the problem of designing a mechanism to implement an efficient or approximately efficient allocation when agents have private information about their preferences. The mechanism design literature has usually restricted attention to mechanisms in which agents fully reveal their preferences, and ensured agents' incentives for truthful revelation (e.g., using the Vickrey-Groves-Clarke transfers to implement efficient allocations). However, full revelation of one's preferences would often be prohibitive: for example, in a combinatorial auction of m items in which agents may have general valuations for the bundles, full revelation requires naming a willingness to pay for each of the $2^m - 1$ bundles. Already with $m = 30$, this would involve the communication of more than one billion numbers.

Because of this communication problem, most real-life mechanisms do not require full revelation. Instead, many mechanisms are "market-like:" they quote to the agents price lists for the allocations (with prices sometimes allowed to be nonlinear and personalized) and request them to submit demands given the prices, adjusting the prices according to some prespecified rules. Indeed, nearly all iterative combinatorial auctions suggested in the literature are based on such "demand queries" (e.g., Parkes and Ungar 2000; Ausubel and Milgrom 2002, and the forthcoming survey of Parkes).

*Noam Nisan's work was supported by grants from the Israeli Science Foundation and the USA-Israel Bi-national Science Foundation. Ilya Segal's work was supported by the U.S. National Science Foundation (grants SES 214500 and ITR 0427770).

†Institute of Computer Science, Hebrew University of Jerusalem, Jerusalem 91904, Israel; email noam@huji.ac.il

‡Department of Economics, Stanford University, Stanford, CA 94305-6072, United States; email ilya.segal@stanford.edu.

The ubiquity of demand-query mechanisms raises the question: Can we restrict attention to mechanisms of this form without blowing up the communication burden? Nisan and Segal (forthcoming) answered this question in the affirmative for the *nondeterministic* problem of verifying the efficiency of an allocation: for this problem, without increasing the communication burden, one can restrict attention to announcing a price equilibrium.[1] But does there exist a parallel theorem for deterministic communication?[2] The goal of this note is to show that the parallel deterministic conjecture is in fact not true. We do this by demonstrating a class of valuations for which the restriction to "demand-query mechanisms" brings about an exponential blowup in the communication burden of finding an efficient allocation. Namely, for this class, an efficient mechanism exists that uses a number of bits that is linear the number of items, but any demand query mechanism that achieves efficiency, or even any improvement upon the "dictatorial" allocation of all the items to one agent, must use an exponential number of demand queries.[3] We also show a parallel average-case result: We construct a joint probability distribution over the agents' valuations from this class for which any improvement in the expected surplus over the dictatorial allocation requires using an exponential expected number of demand queries.

Our results suggest limitations to the usefulness of combinatorial allocation mechanisms such as "iterative auctions," and other "preference elicitation" mechanisms that use demand queries. Allowing "value queries" or "order queries" does not help, since in our setting such queries are easily replicated with demand queries (Conen and Sandholm 2001; Blumrosen and Nisan 2005). Of course, for *some* classes of valuations, or some probability distributions of valuations, demand-query mechanisms may work well. Indeed, we know of several special settings in which a polynomial number of demand queries suffices for obtaining exactly or approximately efficient allocations (e.g., Dobzinski et al. 2005; Lahaie and Parkes 2004; Blum et al. 2004; Nisan and Segal, forthcoming; and the forthcoming survey by Sandholm and Boutilier). However, these cases appear to be quite restricted, and the scope of usefulness of demand-query mechanisms remains unknown.

2 The Valuation Class

Consider the problem of allocating items from a set M between two agents. Let $|M| = m$, which for simplicity we take to be even. Let K denote the set of bundles consisting of $m/2$ items, so $k \equiv |K| = \binom{m}{m/2}$. Each agents' valuation function over bundles of items is known a priori to lie in the class

$$\mathcal{V} = \left\{ \begin{array}{c} v : 2^M \to \{0,1\} \text{ such that} \\ v(S) = 0 \text{ for } |S| < m/2,\, v(S) = 1 \text{ for } |S| > m/2, \\ |\{S \subset K : v(S) = 1\}| > k/2 \end{array} \right\}.$$

An allocation $(S, M \backslash S)$ will be described with the bundle $S \subset M$ of items allocated to agent 1. Each agent i's valuation $v_i \in \mathcal{V}$ can be described by the set of allocations from K for which he

[1] Segal (2004) extends the result to general social choice problems, which may restrict or ban monetary transfers, and which may have goals other than efficiency (e.g., approximate efficiency, fairness, coalitional stability).

[2] This question was raised by Tuomas Sandholm in the 2004 Stanford Institute for Theoretical Economics workshop (oral communication).

[3] We only count the number of demand queries and ignore the problem that each demand query in itself must list prices for an exponential number of bundles. The reason we ignore this problem is that knowledge of the protocol and of the past reports would allow agents to figure out the current prices, and so they need not be described explicitly. For example, in each round of an ascending-price clock auction, the prices for the bundles that have been bid upon are raised by the minimal bid increment and the prices for the other bundles are preserved, and so there is no need to quote all the prices anew in each period.

has value 1, i.e.,

$$\begin{aligned} A_1 &= \{S \subset M : |S| = m/2,\, v_1(S) = 1\}, \\ A_2 &= \{S \subset M : |S| = m/2,\, v_2(M \backslash S) = 1\}. \end{aligned}$$

We will refer to the set $A_i \subset K$ as the "type" of agent i.

Since it is known that $|A_1|, |A_2| > k/2$, we must have $A_1 \cap A_2 \neq \varnothing$, and so a surplus-2 allocation exists. Achieving efficiency is equivalent to finding such an allocation. On the other hand, without finding such an allocation we achieve at most surplus 1, which could have been achieved by giving all the items to agent 1.

Since each agent's type is his private knowledge (subject to commonly-known constraints described above), finding an efficient allocation requires some communication between them. We describe such communication using the model of communication protocols introduced by Yao (1977) and surveyed by Kushilevitz and Nisan (1997).

3 A Fast Efficient Protocol

Proposition 1 *There exists a protocol finding an efficient allocation using no more than* $4(\log_2 k + 1)^2$ *bits of communication.*

Proof. Consider the following protocol: At each step r, we maintain a set $K_r \subset K$ with the property that

$$|A_1 \cap K_r| + |A_2 \cap K_r| > |K_r|, \tag{*}$$

and therefore $A_1 \cap A_2 \cap K_r \neq \varnothing$. Initialize with $K_0 = K$, which satisfies (*) due to the assumption that $|A_1|, |A_2| > |K|$. At each step r, partition K_r (in an arbitrary but prespecified way) into two subsets B_1 and B_2 such that $||B_1| - |B_2|| \leq 1$, and ask each agent $i = 1, 2$ to report $a_{ij} = |A_i \cap B_j|$ for each $j = 1, 2$. Then take $K_{r+1} = B_j$ for the $j \in \{1, 2\}$ that has the higher value of $a_{1j} + a_{2j} - |B_j|$, which guarantees that K_{r+1} satisfies (*). Each step requires communicating no more than $4 \lceil \log_2 |K| \rceil$ bits, and in no more than $\lceil \log_2 |K| \rceil$ steps K_r becomes a singleton, which by (*) must lie in $A_1 \cap A_2$. ■

In fact, there exists an efficient protocol that uses no more than $5.3 \log_2 k$ bits of communication. This follows from the fact that this communication problem turns out to be equivalent to the monotone depth of the majority function (Karchmer and Wigderson, 1988), and Valiant's (1984) celebrated construction of such formulae.

4 Demand-Query Protocols

Now we restrict attention to *demand-query protocols,* which work as follows: at each step, one of the agents is offered a price vector $p : 2^M \to \mathbb{R}$ and an ordering τ over 2^M, both of which can be functions of the agents' previous messages, and asked to report the first element of $\arg\max_{S \subset M} (v_i(S) - p(S))$ in ordering τ. Note the importance of fixing a tie-breaking ordering τ in advance: If agent i's tie-breaking were allowed to depend directly on his valuation v_i, then his choice from a known tie could communicate arbitrary information about v_i, and so the restriction to demand-query protocols would be vacuous.

Another often-used type of query is a "value query," which asks an agent i to report his valuation $v_i(S)$ for a given bundle $S \subset M$. Note that in our simple model such a query is equivalent to a demand query with prices $p(S) = 0$ and $p(T) = 1/2$ for all $T \neq S$ (S will be demanded if and only if $v_i(S) = 1$). Thus, lower communication bounds for demand-query protocols will also apply to protocols that use value queries.

5 The Worst-Case Result

Proposition 2 *Any demand-query protocol that achieves a higher surplus than that from giving all items to one agent must ask at least $k/2 - 1$ queries (in the worst case).*

Proof. Take any protocol that uses less than $k/2 - 1$ demand queries. We describe an "adversary algorithm" for answering a sequence of queries made by the protocol and then constructing agents valuations (described by their types A_1, A_2) consistent with all the answers, for which the total surplus for the protocol's outputed allocation is at most 1. (While constructing the valuations *after* the queries have been made seems like "cheating," the point is that they *could have* been the true valuations from the outset.)

The adversary algorithm at each step maintains sets $B_1, B_2 \subset K$ (with the interpretation that B_i is the set of allocations from K for which agent i "could still have" value 1). The two sets are initialized with $B_1 = B_2 = K$. At each step of the protocol, if agent $i = 1, 2$ is queried, the adversary returns an allocation $a \in 2^M$ that would be demanded by the agent if his type were B_i. (Demanding allocation $a \in 2^M$ by agent i means demanding bundle a if $i = 1$ and demanding bundle M/a if $i = 2$.) Furthermore, if $a \in B_i$, then a is removed from B_{-i}. Proceed to the next step. Suppose that the protocol ultimately outputs allocation t. Then, for each $i = 1, 2$, if $t \in B_i$, t is removed from B_{-i}. Finally, the adversary sets the agents' types to be $A_i = B_i$ for $i = 1, 2$.

Since the protocol has fewer than $k/2 - 1$ steps, at each of which each $|B_i|$ is reduced by at most 1, and the outputed allocation reduces $|B_i|$ by at most 1, in the end we have $|A_i| > k/2$ for $i = 1, 2$, so the constructed types are feasible. Furthermore, by construction we have $t \notin A_1 \cap A_2$. What remains to show is that the demands returned by the adversary at each stage are consistent with the constructed types A_1, A_2. To see this, note first that for each agent i, the sets B_i are nonincreasing in the course of the protocol and thus at each stage, $A_i \subset B_i$. This implies that if at some stage an allocation $a \notin B_i$ was demanded by type B_i, it will also be demanded by type A_i at the same prices and the same tie-breaking rule. On the other hand, if an allocation $a \in B_i$ was demanded by type B_i, then it was removed from B_{-i}, and so by construction it always remains in B_i. Then $a \in A_i$ and so a will also be demanded by type A_i at the same prices and the same tie-breaking rule. Hence, the constructed type A_i will indeed induce the same demands as those constructed by the adversary. ■

Since by Stirling's formula, $k = \binom{m}{m/2} \sim \sqrt{2/(\pi m)} \cdot 2^m$ as $m \to \infty$, the Proposition means that any demand-query protocol improving upon giving all items to one agent requires an exponential number of queries in m. Contrast this to the finding of Proposition 1, which exhibited a protocol achieving efficiency using $O(m^2)$ bits (and in fact, as argued in Section 3, an efficient $O(m)$ protocol exist).

Remark 1 *A matching upper bound follows from Nisan and Segal's (forthcoming) Proposition 1, which implies that the deterministic communication blowup due to a restriction to demand queries is* at most *exponential in any allocation problem with n agents. Indeed, according to their result, each leaf (terminal node) of an efficient protocol must reveal a price equilibrium. If the protocol communicates b bits, then it has at most 2^b leaves. Consider the demand-query protocol that checks* *the candidate price equilibria corresponding to these leaves sequentially, until it finds a true equilibrium, which then yield an efficient allocation. Since a price equilibrium can be checked with n demand queries (one for each agent, with an ordering of allocations in which the candidate equilibrium allocation is the first one), the total number of demand queries used is at most $n2^b$.*

Remark 2 *Conen and Sandholm (2001) introduce "rank queries," which ask an agent to report the bundle of rank r in the order of his valuations, with ties broken according to some a priori ordering τ. Such queries prove more powerful than demand queries for our valuation class. For example, such a query reveals whether the first q bundles in ordering τ contain at least r value-1 bundles. Using bisection on r between 1 and q, we find the number of value-1 bundles with among the first q bundles with at most $\lceil \log_2 q \rceil$ rank queries. Then we can use the protocol described in Proposition 1 doing this at each step, and so the total number of rank queries is at most $4\left(\log_2 k+1\right)^3$. On the other hand, for other valuation classes, rank queries may not achieve efficiency at all, because they do not elicit the strength of the agents' preferences: For example, with only one item, an agent would always rank the item higher than not having it, and we would never learn which agent has the highest valuation for the item.*

6 An Average-Case Result

We start by defining a joint probability distribution D over valuation pairs from $\mathcal{V} \times \mathcal{V}$, by drawing the type pairs uniformly from the pairs (A_1, A_2) such that $|A_1| = |A_2| = k/2+1$ and $|A_1 \cap A_2| = 2$.[4]

Proposition 3 *If the agents' valuation pairs are jointly distributed according to D, any demand-query protocol obtaining an expected surplus of at least $1+\delta$ with $\delta \in (0,1)$ must use at least $T(\delta) = (\delta k/10 - 1/4)/\ln k$ queries in the worst case, and at least $(\delta/2)\,T(\delta/2)$ queries in the average case.*

Proof. Consider first a demand-query protocol asking at most t queries in the worst case. We allow agents to reveal more information than just the demanded bundle. For a given demand query $\langle p, \tau \rangle$, define an ordering π of allocations that agrees with the ascending order of prices p, with ties broken according to τ. When agent i is asked such a query, let him reveal his valuations for the top r allocations from K in ordering π out of those that have not been revealed yet, and let the other agent reveal his valuations for the same allocations (where $r \geq 1$ is a fixed integer). Furthermore, if agent i has value zero for all the allocations he reveals, let him say "bingo," and then let both agents reveal all their valuations. This response is more informative than answering the demand query: if agent i did not say "bingo" and so his valuations are not fully revealed, his demand was given by the highest value-1 allocation in ordering π, which is either among the revealed ones or is a known allocation outside of K.

The probability that an agent said "bingo" in response to one of the t demand queries is bounded above by $t\left(\frac{k/2}{k-rt}\right)^r$, since the fraction bounds above the proportion of either agent's zero-value allocations among those in K that have not been revealed, and for "bingo" we need all the top

[4] It is important for our result that the two agents' valuations are jointly distributed. If the agents' types were drawn independently from a uniform distribution over $A_i \subset K$ such that $|A_i| = k/2+1$, we could achieve efficiency with a small expected number of demand queries, by announcing allocations from K in any fixed order and stopping as soon as we find an efficient allocation (which is verified with two value queries, which are equivalent to demand queries). The probability that a given allocation has value 1 to a given agent is at least 1/2 (higher when more allocations have been checked), and so it is efficient with probability at least 1/4. Thus, the expected number of allocations that need to be checked before finding efficiency is at most 4, and so the expected number of demand queries is at most 8 (regardless of k).

r allocations in K ordering π to be zero-value allocations. Also, if "bingo" has not been said, the probability that the allocation outputed by the protocol is efficient is bounded above by $(rt+1)\,\frac{2}{k}$, since this is the probability that at least one of the two efficient allocations is either one of the rt allocations revealed by the agents or is some other allocation from K outputed by the protocol. Thus, the probability of finding an efficient allocation is bounded above by

$$t\left(\frac{k/2}{k-rt}\right)^{r}+(rt+1)\,\frac{2}{k}$$

We can choose any integer $r \geq 1$, and we choose $r = \lfloor 3\ln k \rfloor$ (to roughly minimize the above expression). Then, if $t \leq T(\delta)$, the probability of efficiency is bounded above by

$$\begin{aligned} & \frac{\delta k/10-1/4}{\ln k}\left(\frac{1}{2-3\delta/5}\right)^{3\ln k-1}+(3\delta k/10-3/4+1)\,\frac{2}{k} \\ \leq\; & \frac{\delta k/10-1/4}{\ln k}\,\frac{2}{k^{3\ln(7/5)}}+(3\delta k/10+1/4)\,\frac{2}{k} \\ \leq\; & \frac{2}{k}\,(\delta k/10-1/4)+\frac{2}{k}\,(3\delta k/10+1/4)\leq 8\delta/10<\delta. \end{aligned}$$

Now suppose that we have a demand-query protocol with an *expected* number t of queries that finds efficiency with probability δ. Then we can terminate the protocol after $2t/\delta$ queries (and in this case output a random allocation from K). The probability that the protocol is terminated is at most $\delta/2$, and so we still have a protocol that finds efficiency with probability $\delta/2$ using at most $2t/\delta$ queries in the worst case. By the previous result, $2t/\delta \geq T(\delta/2)$, and so $t \geq (\delta/2)\,T(\delta/2)$. ■

Remark 3 *If we restricted our valuation class to have* $|A_i| \geq 2k/3$ *for* $i = 1, 2$, *we could always achieve efficiency with a small expected number of demand queries. Indeed, consider the randomized protocol that picks an allocation from* K *uniformly at random and stops as soon as it finds an efficient allocation (which is verified with two value queries, which are equivalent to demand queries). Since there are at least* $k/3$ *efficient allocations, the probability of finding one in each step is at least* $1/3$, *hence the expected number of queries before stopping is at most* 6 *(regardless of* k*). By the Minimax Theorem, this also implies that for every probability distribution on such valuation pairs, there exists an efficient* deterministic *demand-query protocol whose expected number of queries is 6.On the other hand, Proposition 2 is easily extended to this case to show that any demand-query protocol that achieves a surplus greater than 1 must still ask at least* $k/3$ *queries* in the worst case. *Thus, in this case we obtain an exponential divergence between the average-case and worst-case communication complexity of achieving efficiency with demand-query protocols.*

7 Conclusion

We have given a simple example in which a restriction to demand queries brings about an exponential blowup in the communication required to achieve or approximate efficiency. A natural direction for further research is to characterize the valuation classes for which this does not happen. Another interesting question is whether there exists a sufficiently concise class of "universal queries" to which we can restrict attention without causing an exponential communication blowup on *any* valuation class. The Proposition 1 of Nisan and Segal (forthcoming) implies that demand queries do form a universal class for nondeterministic communication, but a parallel question for deterministic communication remains open. Finding a universal query class would be useful for designing practical deterministic mechanisms, while proving that it does not exist would suggest that practical mechanisms should be very dependent on the problem at hand.

References

[1] L. M. Ausubel and P. R. Milgrom. Ascending auctions with package bidding. In *Frontiers of Theoretical Economics*, 1, pages 1-42, 2002.

[2] A. Blum, J. C. Jackson, T. Sandholm and M. A. Zinkevich. Preference elicitation and query learning. In *Journal of Machine Learning Research*, 5, pages 649-667, 2004.

[3] L. Blumrosen and N. Nisan. On the computational power of iterative auctions. In *Proceedings of the ACM Conference on Electronic Commerce*, 2005.

[4] W. Conen and T. Sandholm. Preference elicitation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce*, 2001.

[5] S. Dobzinski, N. Nisan and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of the ACM Symposium on Theory of Computing*, 2005.

[6] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the 20th ACM Symposium on Theory of Computing*, pages 539–550, 1988.

[7] E. Kushilevitz and N. Nisan. *Communication Complexity.* New York: Cambridge University Press, 1997.

[8] S. Lahaie and D. C. Parkes. Applying learning algorithms to preference elicitation. In *Proceedings of the ACM Conference on Electronic Commerce*, 2004.

[9] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, forthcoming.

[10] D.C. Parkes. Iterative combinatorial auctions. Chapter 3 in P. Cramton, R. Steinberg, and Y. Shoham, *Handbook on Combinatorial Auctions*, MIT Press, forthcoming.

[11] D. C. Parkes and L. H. Ungar. Iterative combinatorial auctions: theory and practice. In *AAAI/IAAI*, pages 74-81, 2000.

[12] T. Sandholm and C. Boutilier. Preference elicitation in combinatorial auctions. Chapter 10 in P. Cramton, R. Steinberg, and Y. Shoham, *Handbook on Combinatorial Auctions*, MIT Press, forthcoming.

[13] Segal, I. The communication requirements of social choice rules and supporting budget sets, working paper, Stanford University, 2004.

[14] L. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms* 5, pages 363–366, 1984.

[15] A. C. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proceedings of the 17th Annual Symposium on Foundations of Computer Science*, pages 222–227, 1977.

[16] A.C. Yao. Some complexity questions related to distributed computing," *Proceedings of the 11th ACM Symposium on the Theory of Computing*, pages 209-213, 1979.