CORE DISCUSSION PAPER

2005/87

MECHANISM DESIGN FOR MULTIPLE ITEM PROCUREMENT USING A DISTRIBUTED ELLIPSOID ALGORITHM

Andrew J. Miller[1], Debasis Mishra[2] and Dharmaraj Veeramani[3]

December 2005

Abstract

We study a problem in a procurement setting in which an *Original Equipment Manufacturer* (OEM) wants to procure a set of items from a set of suppliers. Each supplier incurs a cost for supplying any subset/bundle of items, and each supplier's cost information is known only to him. The goal is to determine (i) an *efficient* allocation of suppliers to items and (ii) appropriate payments for suppliers. We formulate the problem of determining which supplier should supply what items as an integer program. We develop a method, which is based on the *ellipsoid* method, that solves the dual of the linear relaxation of the problem in polynomial time. We also show that the linear relaxation has an integral optimal solution.

[1]Department of Industrial and Systems Engineering, University of Wisconsin-Madison, USA.
[2]Center for Operations Research and Econometrics (CORE), Belgium.
[3]Department of Industrial and Systems Engineering, University of Wisconsin-Madison, USA.

The research of Andrew J. Miller is supported by NSF grant DMII-0323299. The authors thank Kent Andersen for his feedback on the work.
This text presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

1 Introduction

Recently a large number of companies have moved to Internet enabled procurement. With the growth of the Internet, the use of reverse auctions on the Internet for procurement has also increased. Often, companies need to procure a set of items simultaneously, but in current practice each item is procured separately. This leads to loss in efficiency and higher costs of procurement. In this paper, we seek to answer the following question. How can a set of items be procured simultaneously and efficiently from a set of suppliers?"

The model considered involves a customer or an *Original Equipment Manufacturer* (OEM) who wants to procure a set of n distinct heterogeneous items. In the context of this paper, an item is any indivisible object. By "indivisible", we mean that the object needs to be procured as one item, even though the actual object can be a collection of entities (like a lot of metal sheets, for example). There are m suppliers who can supply the items. Each supplier incurs a cost for supplying any subset (or *bundle*) of items. We are interested in answering the following two questions:

(Q1) Which supplier should supply what items?

(Q2) What prices should the suppliers be paid?

In this paper, we attempt to design a *mechanism* that answers these two questions. We aim at a mechanism that ensures that the suppliers report their true costs. We call this the *procurement economy problem*. The main difficulty is that the suppliers may not maximize their individual utilities by reporting their true costs. In mechanism design theory, suppliers are paid an amount that ensures that they have no incentive to misreport their costs [1]. We apply the principle of mechanism design to answer (Q1) and (Q2). To answer (Q1), we formulate the problem as an Integer Program (IP). The objective is to minimize the total cost of supplying the items. We show that under certain conditions, the linear relaxation of the IP gives an optimal solution of the IP. To solve the linear program, we develop an algorithm that is based on the famous ellipsoid method for linear programming. To answer (Q2), we consider two pricing schemes. One is a pricing scheme from economics known as the Walrasian equilibrium pricing. The other is the well-known pricing scheme from mechanism design called the Vickrey-Clarke-Groves (VCG) mechanism [23, 4, 9]. In the VCG mechanism, allocation is done *optimally* (by solving the underlying optimization problem) and agents are paid in such a way that their net utility is equal to their marginal contribution to the system. Such a pricing scheme makes submission of true costs by suppliers a (weakly) dominant strategy. We propose an algorithm to compute Walrasian equilibrium prices and the VCG payments.

[1] Readers not familiar with mechanism design theory are suggested to read [12] for a quick introduction.

1.1 Related Work

Most of the prior research related to our work is done in a *buyers-seller* setting. In a buyers-seller setting/economy, a seller is selling a set of items to a set of buyers who each have values on subsets of items (bundles). The question in that case becomes who should be assigned which item and at what price. The buyer-seller setting is a natural counterpart of our setting, and many results of that setting can also be derived for our setting.

The buyers-seller problem has been studied in a simplified setting where each buyer wants to buy at most one item. The concept of the VCG mechanism can be directly applied to such a setting. Leonard [16] discusses such a pricing scheme in the setting of labor markets. In a labor market, a set of positions need to be filled by a set of individuals who have private valuations on positions. He shows that the VCG prices in such a setting constitute the dual solution of a linear programming assignment problem. He further shows that such prices constitute the *Walrasian equilibrium price*. Later, Demange et al. [7] proposed two ascending-price auctions for such a setting. One of them achieves exact VCG prices and the other achieves them approximately. The advantages of these mechanisms are that they can be implemented in a distributed manner and do not require buyers to report their valuations *directly*.

The results of the unit demand case in a buyers-seller setting have been extended (to some extent) to the case of multiple item demands. Kelso and Crawford [13] introduced a condition, called the *gross substitutes* (GS) condition, under which Walrasian equilibrium will exist in such an economy. Gul and Stacchetti [10, 11] have two papers in this setting. In their earlier paper [10], they give equivalent conditions to the GS condition and show that the Walrasian equilibrium price vectors form a lattice. In the same paper they show that the payment by a buyer in any Walrasian equilibrium pricing scheme is always at least his payment in the VCG mechanism. They show that this *gap* vanishes for large economies. In their second paper [11], they show a strong relationship between their work and matroid theory. They propose an ascending-price auction which converges to the minimum Walrasian equilibrium price vector. They give conditions under which this auction inherits the good incentive properties of the VCG mechanism. Bikhchandani et al. [3] model this problem as an integer program (IP). They give various strong formulations and show that under a condition called *agents are substitutes*, one of the formulations has dual variables which relate to the VCG prices. They also give an IP formulation that has an integral optimal solution if the GS condition (or equivalently any of the conditions in [10]) hold. They propose primal-dual based algorithms to discover these prices. Parkes [20] proposes an auction which achieves the VCG outcome by maintaining personalized prices for buyers and prices for every bundle. Ausubel [1] has also proposed an iterative mechanism which discovers the VCG prices. Recent work by Ausubel and Milgrom [2] and de Vries et al. [6] propose iterative auctions under these models and give conditions under which these auctions achieve the VCG outcome.

Finding the VCG payments is an NP-Complete problem in a multiple item demand case (see for example [21]). Therefore much research has been done in addressing the computational complexity issue. Some of the research that discusses this issue includes [20, 3, 15, 22]. Most of this research is in developing mechanisms/algorithms of distributed nature or in developing efficient *message protocols* to elicit agents' preferences to address the computational complexity issue.

Comparing our contribution with the research in buyers-seller setting, we observe that the fundamental optimization problem in the procurement economy is different. We are concerned with finding an equilibrium price vector in our procurement setting. Bikhchandani et al. [3] related a linear programming formulation and the gross substitutes property in a buyers-seller setting. Similarly, in the buyers-seller setting, under the gross substitutes condition, it is known that equilibrium prices can be found with polynomial number of calls to the *demand oracle* [19]. In [17], the demand oracle is shown to be solvable in polynomial time for a very specific type of valuation function of the buyers. More recently, Murota and Tamura [18] gave a polynomial time algorithm to find equilibrium prices under gross substitutes condition. Their algorithm relates concepts from discrete convexity to gross substitutes condition (also see Danilov et al. [5] for connection of discrete convexity to the gross substitutes condition).

1.2 Contribution of this Paper

In this paper, we answer some questions relating to the procurement economy problem. We formulate the problem as an integer program (IP). We first show that if cost functions are monotonic and satisfy a condition called *no production complementarity* (NPC), introduced in Gul and Stacchetti [10], then the linear relaxation of IP has an integral optimal solution [2]. Moreover, we demonstrate that the dual variables of this LP relaxation corresponds to the Walrasian equilibrium prices at optimality. This result is similar to earlier results in the literature on the buyers-seller setting [3].

These results allow us to propose a technique based on the ellipsoid method to solve the dual problem. The ellipsoid method solves the *separation problem* (that is, it either certifies that a given vector is dual feasible, or it returns a violated dual inequality) a polynomial

[2]We will elaborate more on the NPC condition later.

number of times. To solve the separation problem, in our setting, turns out to consist of asking each supplier whether or not the dual vector satisfies all the dual constraints of the supplier, and if not, to have the supplier deliver a dual constraint that is violated the most. Even though there is an exponential number of dual constraints associated to each supplier, the fact that the ellipsoid method only requires the solution of a polynomial number of separation problems ensures that the overall method runs in polynomial time.

The method described above runs in a distributed manner. We prove that the VCG prices paid to suppliers will always be (weakly) greater than the price that needs to be paid in a Walrasian equilibrium pricing scheme. We show how to compute the maximum Walrasian equilibrium prices and the VCG prices using the ellipsoid algorithm. We also discuss incentive issues. This technique is non-conventional and it has some interesting practical properties. It has nice properties of iterative mechanisms like reverse auctions as it is distributed in nature and works on prices. Also, it carries the nice properties of sealed-bid auctions (RFQ based procurement) in the sense that it reveals little information to the public.

The rest of the paper is organized as follows. In Section 2, we formally define the problem. In Section 3, we introduce the concept of Walrasian equilibrium and formulate the problem as an integer program. In Section 4, we propose a method based on ellipsoid algorithm to discover Walrasian equilibrium prices. Section 5 describes when the ellipsoid algorithm is incentive compatible for suppliers. We summarize and describe future research directions in Section 6.

2 Problem Statement

We assume there is a *customer* agent who needs to procure a set of n heterogeneous items. Denote this set of items by $A = \{1, 2, \ldots, n\}$. There is a set of m suppliers that can supply each subset of these items. Let $B = \{1, 2, \ldots, m\}$ be the set of suppliers. Let Ω be the power set of A; so $\Omega = \{S : S \subseteq A\}$. Each supplier has a cost of supplying a bundle $S \in \Omega$. We denote the cost of supplying $S \in \Omega$ by supplier $i \in B$ as $c_i(S)$. We assume $c_i(\emptyset) = 0 \; \forall \; i \in B$. If a supplier does not have the capability to supply a bundle of items, the cost on that bundle can be set to a very high amount. We denote the vector of all prices by $p \in \mathbb{R}^A_+$, and the price of a single item $j \in A$ by p_j. If a supplier i supplies a set of items S at price p, then his *payoff* or *utility* is $u_i(S, p) = \sum_{j \in S} p_j - c_i(S)$. We assume the cost function for each supplier is monotonically increasing, i.e., for any supplier $i \in B$, we have $c_i(S) \leq c_i(T) \; \forall \; S \subseteq T \in \Omega$.

Definition 1 *The* supply set *at price vector p for a cost function c is defined as $L(p) =$*

$\{S : u(S, p) \geq u(T, p) \ \forall \ T \in \Omega\}$.

Supply set contains the utility-maximizing set of bundles of items at a price vector. Using the notion of supply set, we give some further definitions.

Definition 2 *A cost function $c : \Omega \to \mathbb{R}_+$*
(i) satisfies Gross Substitutes for Production (GSP) *condition, if for any two price vectors p and q with $p \geq q$, and any $S \in L(q)$, there exists a $T \in L(p)$ such that $\{j \in (A \setminus S) : p_j = q_j\} \subseteq (A \setminus T)$.*
(ii) satisfies the Single Improvement (SI) *property, if for any price vector p and any bundles $S \notin L(p)$ there exists a bundle T such that $u(T, p) > u(S, p)$ and $|S \setminus T|, |T \setminus S| \leq 1$.*
(iii) has No Production Complementarity (NPC)*, if for every price vector p, $S, T \in L(p)$ and $Y \subseteq S \setminus T$, there exists a $Z \subseteq T \setminus S$, such that $(S \setminus Y) \cup Z \in L(p)$.*

The NPC and SI conditions were introduced in [10]. The GS condition was introduced in a buyers-seller economy by Kelso and Crawford [13]. We modify their condition to the procurement setting. By the arguments in Section 6 and Theorem 1 of [10], if c is monotone, then GSP, SI and NPC are equivalent.

These conditions, to a large extent, preclude the existence of items which are complements of each other. For example, the SI condition says that we can improve a bundle not in the supply set by either adding an item or deleting a item or doing both.

We assume that the customer has a reservation cost that indicates the maximum amount the customer is willing to pay for a given bundle. We implement this idea using a dummy supplier, 0. The dummy supplier has a cost function c_0. We assume additive costs for the dummy supplier, i.e., there are costs on individual items and the cost on a bundle $S \in \Omega$ is simply $c_0(S) = \sum_{j \in S} c_{0j}$, where c_{0j} is the cost on item j. Thus the prices on items are bounded above by the cost of the dummy supplier. We will denote $B_0 = B \cup \{0\}$ and $B_{-i} = B \setminus \{i\}$.

3 Walrasian Equilibrium

The economy $E = (A; c_0, c_1, \ldots, c_m)$ consists of a finite collection of items, A, and the set of suppliers B_0 with their cost functions. The procurement problem can be formulated as the following integer program (**P**).

$$\min \sum_{i \in B_0} \sum_{S \in \Omega} c_i(S) y(i, S)$$

s.t.

$$\sum_{S\in\Omega} y(i,S) \leq 1 \;\forall\; i \in B$$

$$\sum_{S\in\Omega: j\in S}\sum_{i\in B_0} y(i,S) = 1 \;\forall\; j \in A$$

$$y(i,S) = \{0,1\} \;\forall\; i \in B, \;\forall\; S \in \Omega$$

The first set of constraints of **P** say that each supplier in B must be assigned at most one subset of items. The second set of constraints say that each item must be assigned exactly once. It is clear that a feasible solution of **P** is a partitioning $X = (X_1, \ldots, X_m)$ such that $y(i, X_i) = 1 \;\forall\; i \in B$ and $y(i,S) = 0$ if $X_i \neq S \;\forall\; i \in B, \;\forall\; S \in \Omega$.

Gul and Stacchetti [10] discuss Walrasian equilibria in such a setting. The Walrasian equilibrium in such a setting can be defined as follows:

Definition 3 *Let (p, X) be a tuple, where $p \in \boldsymbol{R}^n_+$ is a price vector and $X = (X_0, X_1, \ldots, X_m)$ is a partition of A with X_i representing the allocation of supplier $i \in B$. (p, X) is a Walrasian equilibrium of the economy $E = (A; c_0, \ldots, c_m)$ if*

- *for every $i \in B$, $X_i \in L_i(p)$.*
- *$p_j \leq c_{0j} \;\forall\; j \in A$ and if $j \in X_0$, $p_j = c_{0j}$.*

The price vector p is called the Walrasian equilibrium price vector.

Denote $A_0(p) = \{j \in A : p_j = c_{0j}\}$. We can write the Walrasian equilibrium as a solution (p, y) to the following set of equations (**CE**):

$$\begin{aligned}
\sum_{S\in L_i(p)} y(i,S) &= 1 \qquad \forall\; i \in B \\
\sum_{S\ni j}\sum_{i\in B} y(i,S) &= 1 \qquad \forall\; j \notin A_0(p) \\
\sum_{S\ni j}\sum_{i\in B} y(i,S) &\leq 1 \qquad \forall\; j \in A_0(p) \\
y(i,S) &\in \{0,1\} \qquad \forall\; i \in B, \;\forall\; S \in \Omega
\end{aligned}$$

By Theorem 10 of [10], if c_i is monotone and satisfies NPC/SI/GSP for every supplier $i \in B$, then the Walrasian equilibrium of the economy $E = (A; c_0, \ldots, c_m)$ exists and the set of Walrasian equilibrium prices form a complete lattice. Using this result, we can prove the following theorem. A parallel result for the buyers-seller setting appears in [3].

Theorem 1 *If each supplier's cost function is monotone and satisfies SI, then there exists an optimal solution of* ***LP*** *which is an optimal solution of* ***P****.*

Proof: If each supplier's cost function is monotone and satisfies SI, then Walrasian equilibrium exists [10]. We can find a (p, y) which is a Walrasian equilibrium. All the items $j \in A_0(p)$ can be assigned to the dummy supplier. In particular define $X_0 = \{j \in A_0(p) : \sum_{S \ni j} \sum_{i \in B} y(i, S) = 0\}$ and let $y(0, X_0) = 1$. Now, y is feasible solution of **LP**. Consider the dual of **LP**: (**DP**)

$$\max_{p,\pi} \sum_{j \in A} p_j - \sum_{i \in B} \pi_i$$

s.t.

$$\sum_{j \in S} p_j \leq \sum_{j \in S} c_{0j} \qquad \forall\ S \in \Omega \tag{1}$$

$$\sum_{j \in S} p_j - \pi_i \leq c_i(S) \qquad \forall\ i \in B,\ \forall\ S \in \Omega \tag{2}$$

$$\pi_i \geq 0 \qquad \forall\ i \in B$$

If (p, y) is a Walrasian equilibrium with $p_j \geq 0\ \forall\ j \in A$, define for every $i \in B$, $\pi_i = u_i(S, p) = \sum_{j \in S} p_j - c_i(S)$, where $S \in L_i(p)$. This means that (p, π) is a feasible solution to **DP**. Now consider the complementary slackness (CS) conditions:

CS1 $y(0, S)(\sum_{j \in S}[c_{0j} - p_j]) = 0\ \forall\ S \in \Omega$.

CS2 $y(i, S)(\pi_i - [\sum_{j \in S} p_j - c_i(S)]) = 0\ \forall\ i \in B,\ \forall\ S \in \Omega$.

CS3 $\pi_i(1 - \sum_{S \in \Omega} y(i, S)) = 0\ \forall\ i \in B$.

CS1 is satisfied because, $p_j \leq c_{0j}$ and if $y(0, S) = 1$, then $S = X_0$ and $p_j = c_{0j}\ \forall\ j \in X_0$. CS2 is satisfied because, if $y(i, S) = 1$ then $S \in L_i(p)$. This means, $\pi_i = \sum_{j \in S} p_j - c_i(S)$. CS3 is satisfied because of feasibility of **CE** and the first set of constraints in **CE**. This means (p, π) is dual feasible, y is primal feasible and (p, π, y) satisfies CS conditions. By duality theory, y is an optimal solution of **P**. ■

4 Distributed Ellipsoid Algorithm

To solve a linear program in polynomial time, a method known as the *ellipsoid algorithm* was proposed by Khachiyan [14]. Grötschel, Lovász, and Schrijver [8] later showed that the ellipsoid algorithm can solve a linear program in polynomial time if and only if a given point can be separated from the feasible polyhedron in polynomial time. The problem of separating a point is also called the *separation problem* [3]. We will apply this idea to solve **DP** (the dual of **LP**).

For the sake of brevity, we omit a detailed description of the general ellipsoid algorithm. We now give a brief summary of how it would solve a linear program like **DP**. An initial ellipsoid is defined so as to contain the entire feasible region. Starting from this initial ellipsoid, the method can be seen as constructing a sequence of ellipsoids such that each contains both the set of optimal solutions and one half of the volume of the preceding ellipsoid in the sequence. The final ellipsoid will be small enough so that its center will be an optimal solution, or close enough that it can be converted to one in polynomial time.

The important point for our discussion is that it can be shown that the number of ellipsoids in the sequence is bounded by a polynomial in the number of variables (here, m and n). Moreover, at each stage, the next ellipsoid in the sequence is constructed in part by determining whether or not the center $(\hat{p}, \hat{\pi})$ of the current ellipsoid is a dual feasible point, and (if not) identifying a violated dual inequality. This separation problem is the bottleneck in defining the next ellipsoid, and therefore if this can be done in polynomial time, each iteration of the ellipsoid can be define in polynomial time. Thus, although **DP** has an exponential number of constraints, it is possible to solve it using an ellipsoid algorithm in polynomial time if and only there is a polynomial time algorithm for the separation problem.

Observe that, given a vector $(\hat{p}, \hat{\pi})$, all constraints of the form (1) will be satisfied if and only if each of the n constraints $p_j \leq c_{0j}$, $\forall\, j \in A$, are satisfied. Finding a violated inequality of the form (1), or showing that there are none violated, is therefore trivial.

Our discussion of the separation problem for **DP** will therefore focus on the constraints (2). We will state the separation problem as follows:

- For every supplier $i \in B$; either
  - show that $\sum_{j \in S} \hat{p}_j - \hat{\pi}_i \leq c_i(S)\ \forall\ S \in \Omega$; or
  - find an $S^* \in \Omega$ for which the constraint $\sum_{j \in S^*} p_j - \pi_i \leq c_i(S^*)$ is violated.

[3] In addition to containing a detailed development of the result just cited, [8] describes many other aspects and consequences of using ellipsoid algorithms to solve combinatorial optimization problems

Thus, to solve the separation problem for **DP**, each supplier solves a *local separation problem*. Observe that for each supplier $i \in B$,

$$\begin{aligned}
\sum_{j \in S} p_j - \pi_i &\leq c_i(S) \ \forall \ S \in \Omega \\
\iff \pi_i &\geq \sum_{j \in S} p_j - c_i(S) \ \forall \ S \in \Omega \\
\iff \pi_i &\geq \max_{S \in \Omega} \Big[\sum_{j \in S} p_j - c_i(S) \Big] \\
\iff \pi_i &\geq \max_{S \in \Omega} u_i(S, p).
\end{aligned}$$

This means that, to solve the separation problem, every supplier needs to identify a bundle which gives him the maximum utility (call it an *optimal* bundle). The ellipsoid algorithm will therefore run in polynomial time if and only if, for any given price vector, each supplier can find his optimal bundle in polynomial time.

To identify an optimal bundle for a given price vector p for some supplier $i \in B$, consider the following simple search procedure. Choose any bundle $S \in \Omega$. If S is not an optimal bundle for i, then by single improvement, we can improve S in 3 ways:

- By adding an item to S ($n - |S|$ choices).
- By deleting an item from S ($|S|$ choices).
- By doing both of the above ($|S|(n - |S|)$ choices).

A non-optimal bundle can therefore be improved in $\mathcal{O}(n^2)$ time. We can keep improving non-optimal bundles till we find an optimal bundle. If costs and prices are all integers and costs are polynomial in (m, n) (the number of suppliers and items), then this search process can find an optimal bundle in polynomial time. This gives us a *pseudo-polynomial* time search procedure for the separation problem, and thus the ellipsoid algorithm will run in pseudo-polynomial time. This immediately gives us our next observation.

Observation 1 *Optimization of both **DP** and **LP** can be done in pseudo-polynomial time using the ellipsoid algorithm.*

Murota and Tamura [18] related discrete convexity to this kind of mathematical economics. Using concepts from discrete convexity that can address submodular flow problems effectively, they provide an algorithm to find competitive equilibrium prices in polynomial time. This means that it is possible to solve both **LP** and **DP** in polynomial time using their algorithm.

This also means that it is possible to solve the separation problem for **DP** in polynomial time as well. For example, if $(\hat{p}, \hat{\pi})$ is not optimal for **DP**, it is possible to define a violated constraint by using the algorithm of [18] to find an optimal bundle S for each supplier i. This could be done for each i by running their algorithm on a reduced version of **P** that has only variables related to suppliers i and 0 (the dummy supplier), and for which the costs on these variables are modified to be $-u_i(S,p)$ for i and 0 for supplier 0, for each bundle $S \in \Omega$.

While an important breakthrough, the algorithm in [18] does not have the following two important aspects of the ellipsoid algorithm that we have described: (i) The ellipsoid algorithm is a distributed method in which each supplier solves his own separation problem and submits his violated constraint in each iteration. (ii) The ellipsoid algorithm uses a minimal number of constraints required to solve **DP**, i.e., suppliers only submit constraints that are *necessary* to solve **DP**. This means that suppliers will not reveal costs on all bundles, but only on bundles for which constraints are submitted. This cost information revelation is strategically important in a procurement setting. The algorithm in [18] assumes full knowledge of cost functions of all suppliers.

In practice, it is unlikely that using a distributed ellipsoid algorithm will be practical or desirable. However, it will be interesting to investigate how the following simple procedure performs in practice in terms of number of iterations to converge to an optimal or near-optimal assignment and price vector).

- **Step 0:** Start with an LP with the same set of variables and objective function as **DP**, with constraints $p_j \le c_{0j}\ \forall\ j \in A$, and with non-negativity constraints on the variables. Find a solution vector $(\hat{p}, \hat{\pi})$ to this LP.

- **Step 1:** For every supplier $i \in B$, either

  - show that $\sum_{j \in S} \hat{p}_j - \hat{\pi}_i \le c_i(S)\ \forall\ S \in \Omega$; or
  - find an $S^* \in \Omega$ for which this constraint is most violated. Add $\sum_{j \in S^*} p_j - \pi_i \le c_i(S^*)$ as a cut to the LP.

  (This can be done by the simple single improvement search procedure described above).

- **Step 2:** If no cuts were found in Step 1, we have found a dual optimal solution (p^*, π^*). Using complementary slackness, we can find a feasible solution X to the primal and

STOP. Else, solve the main problem with the added cuts to find the new solution $(\hat{p}, \hat{\pi})$ and go back to Step 1.

5 Incentive Compatibility of Ellipsoid Algorithm

By Gul and Stacchetti [10], the Walrasian equilibrium prices of our economy form a complete lattice. This means there exist unique minimum and maximum Walrasian equilibrium price vectors. The ellipsoid algorithm converges to one of the Walrasian equilibrium price vectors. The question is whether the suppliers can manipulate the algorithm to achieve higher utility at the end of the algorithm. To analyze this, we first discuss a well-known *strategyproof* payment scheme, called the Vickrey-Clarke-Groves (VCG) payment scheme [23, 4, 9]. Under the VCG scheme, a *mechanism* implements an *efficient* (optimal) *decision rule* (allocation) and pays the participating agents their marginal contributions. To understand the VCG payments, in our setting, let $V(X)$ denote the optimal objective value of formulation **P** when a set of suppliers $X \subseteq B$ are considered. $V(B)$ is therefore the optimal objective value when all suppliers are considered and $V(B_{-i})$ is the optimal objective value when all but i are considered. By the VCG scheme, if i is allocated S_i in the optimal allocation, then he should be paid $c_i(S_i) + V(B_{-i}) - V(B)$. (Note that the payments in such a scheme are a map from suppliers to positive real numbers whereas Walrasian equilibrium prices are a map from items to positive real numbers.) The VCG payment of supplier i, assigned to S_i in the optimal allocation of **P**, becomes (denote it as p_i^{vcg}):

$$p_i^{vcg} = c_i(S_i) + V(B_{-i}) - V(B). \tag{3}$$

Now, we can prove a theorem which provides a relationship between Walrasian equilibrium prices and the VCG prices. A parallel theorem in the buyers-seller setting is proved by Gul and Stacchetti [10].

Theorem 2 *If supplier $i \in B$ is allocated S_i in a Walrasian equilibrium allocation, then $p_i^{vcg} \geq \sum_{j \in S_i} p_j$, where p is the corresponding Walrasian equilibrium price vector.*

Proof: If p is a Walrasian equilibrium, then there exists (π, y) such that y is an optimal solution of **P** with $y(i, S_i) = 1$ and (p, π) is an optimal solution of **DP**. By duality, $\pi_i \leq V(B_{-i}) - V(B)$. This is because $V(B_{-i}) - V(B)$ is the value of reducing the right hand side of the corresponding primal constraint. This implies that $p_i^{vcg} = c_i(S_i) + [V(B_{-i}) - V(B)] \geq c_i(S_i) + \pi_i = c_i(S_i) + \sum_{j \in S_i} p_j - c_i(S_i) = \sum_{j \in S_i} p_j$. ■

The theorem proves that the VCG payment of a supplier is at least the maximum Walrasian equilibrium price of the bundle assigned to the supplier. In general, there can be a *gap* between the two types of payments.

5.1 Achieving Maximum Walrasian Equilibrium

The ellipsoid algorithm in the previous section achieves a Walrasian equilibrium. But we did not mention how it can achieve a particular Walrasian equilibrium. Now we prove a theorem which proves the necessity of achieving the maximum Walrasian equilibrium. First we define some notation. Let $\mathbb{C}$ be the class of monotone and GS cost profiles. For the true cost profile, c denotes the maximum Walrasian equilibrium price vector as p^{max}. We denote the true submission of cuts by suppliers in the ellipsoid algorithm a *straightforward* strategy. We say that the straightforward strategy is in *ex post Nash equilibrium* if for every supplier $i \in B$, i can not gain more payoff by following some other strategy when all suppliers other than i follow the straightforward strategy.

Theorem 3 *Let the allocation of supplier i be denoted as S_i. If $p_i^{vcg} = \sum_{\alpha_i \in S_i} p_i^{max}$ $\forall$ $i \in B$ and the ellipsoid algorithm converges to p^{max} (we will give an exact procedure to compute p^{max} later), then the straightforward strategy is an ex post Nash equilibrium for each supplier in every step of the ellipsoid algorithm.*

Proof: Consider a supplier $i \in B$ and his cost function $c_i \in \mathbb{C}$. Suppose supplier i misreports at some iteration of the ellipsoid algorithm. As a result, let him be allocated T. Clearly, him being allocated S_i is weakly preferred to $T = \emptyset$. Now, consider a cost function of i which allocates T in a Walrasian equilibrium of the economy. Specifically, consider the following cost function:

$$\begin{aligned} c_i'(S) &= 0 \text{ if } S \subseteq T \\ &= \sum_{j \in S} p_j \text{ otherwise} \end{aligned}$$

Assigning T to i is a Walrasian equilibrium for this economy. This means that in an optimal allocation of this economy, i can be assigned T. Let $\hat{p}_i^{vcg}$ be the VCG payment of i in such an economy and $\hat{p}$ be the final price vector in the ellipsoid algorithm that supports a Walrasian equilibrium in such an economy. Since the VCG mechanism is strategyproof and assigns items using an optimal allocation, we get

$$\sum_{j \in S_i} p_j^{max} - c_i(S_i) = p_i^{vcg} - c_i(S_i) \geq \hat{p}_i^{vcg} - c_i'(T)$$

By Theorem 2, $\hat{p}_i^{vcg} \geq \sum_{j \in T} \hat{p}_j$. This gives us

$$\sum_{j \in S_i} p_j^{max} - c_i(S_i) \geq \sum_{j \in T} \hat{p}_j - c_i'(T)$$

This means following some other strategy weakly lowers payoff of i. ■

The theorem shows that of all the Walrasian equilibrium price vectors, the maximum Walrasian equilibrium price vector is the closest to achieving an ex post Nash equilibrium. To achieve the maximum Walrasian equilibrium using an ellipsoid algorithm based approach, first run the algorithm described in the previous section to obtain the optimal solution to **DP**, and let its objective function value be V_{OPT}. Now solve the following LP:

$$\max_p \sum_{j \in A} p_j$$

s.t.

$$\begin{aligned} \sum_{j \in S} p_j \leq \sum_{j \in S} c_{0j} & \qquad \forall\ S \in \Omega \\ \sum_{j \in A} p_j - \sum_{i \in B} \pi_i = V_{OPT} & \\ \sum_{j \in S} p_j - \pi_i \leq c_i(S) & \qquad \forall\ i \in B,\ \forall\ S \in \Omega \\ \pi_i \geq 0 & \qquad \forall\ i \in B \end{aligned} \tag{\textbf{MAXCE}}$$

Theorem 4 *The optimal solution of **MAXCE** is the maximum CE price, and it can be computed in polynomial time.*

Proof: Clearly, every feasible solution of **MAXCE** is also a feasible solution of **CE**. **MAXCE** looks for an optimal solution of **CE** which maximizes the sum of prices. Let p^{max} be the maximum CE price and p be the optimal solution of **MAXCE**. By the lattice nature of CE price space, the maximum CE price vector is unique and $\sum_{j \in A} p_j^{max} > \sum_{j \in A} p_j$. This means that $p = p^{max}$.

Also, since the separation problem for **MAXCE** is essentially the same as that for **DP**, we know from the previous section that it can be solved in polynomial time. Thus p^{max} can be calculated in polynomial time. ■

5.2 Calculating the VCG Prices

Since the ellipsoid algorithm runs in polynomial time, we can run it $m+1$ times to compute the VCG prices. The first run of the ellipsoid algorithm involves all the suppliers in B. In each of the subsequent m runs of the algorithm, a different supplier is removed, and the problem is solved for the supplier set B_{-i}. Each of these runs therefore computes the marginal contribution of an agent i. At the end of these $m+1$ runs, we can compute the VCG prices. As proved earlier, behaving honestly at each stage of the ellipsoid algorithms is an ex post Nash equilibrium for the suppliers. Finally, while it may be unlikely that the ellipsoid algorithm will be useful in computing the VCG prices in practical scenarios, it seems likely that a straightforward method like that described at the end of Section 4 would be simple to implement, allow for limited information sharing, and be computationally fast.

6 Summary and Future Research

We formulated the procurement problem as an integer program and showed that under monotonicity and single improvement, its linear relaxation gives an optimal solution. Using this, we propose an ellipsoid algorithm based approach to solve its dual and we show how the dual can be solved in polynomial time. We discuss the incentive compatibility issues of the ellipsoid algorithm. The ellipsoid algorithm can be implemented in a distributed manner and it asks the suppliers to submit the costs of bundles which are *necessary* to solve the problem.

In the presence of monotonicity and GS condition, our ellipsoid algorithm runs in polynomial time. We would like a more straightforward polynomial method that, for example, directly exploits the single improvement property to solve the local separation problem for each supplier. At present, however, we do not know of such an algorithm. We also want to investigate how to apply the ellipsoid algorithm based approach to implement a non-linear and non-anonymous pricing scheme (like the VCG pricing). Also, it will be interesting to explore direct primal-based methods that allow distributed information sharing.

References

[1] L. M. Ausubel. An Efficient Dynamic Auction for Heterogenous Commodities. Working Paper, University of Maryland, Forthcoming in American Economic Review, 2006.

[2] L. M. Ausubel and P. R. Milgrom. Ascending Auctions with Package Bidding. *Frontiers of Theoretical Economics*, 1(1):1–42, 2002.

[3] S. Bikhchandani, S. de Vries, J. Schummer, and R. Vohra. Linear Programming and Vickrey Auctions. In *Mathematics of the Internet: E-Auction and Markets*, volume 127, pages 75–116. IMA Volumes in Mathematics and its Applications, 2002.

[4] E. Clarke. Multipart Pricing of Public Goods. *Public Choice*, 8:19–33, 1971.

[5] V. Danilov, G. Koshevoy, and Lang C. Gross Substitution, Discrete Convexity, and Supermodularity. *Discrete Applied Mathematics*, 131:283–298, 2003.

[6] S. de Vries, J. Schummer, and R. Vohra. On Ascending Vickrey Auctions for Heterogeneous Objects. Working Paper, Northwestern University, Forthcoming in Journal of Economic Theory, 2005.

[7] D. Gabrielle, D. Gale, and M. Sotomayor. Multi-item Auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[8] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.

[9] T. Groves. Incentives in Teams. *Econometrica*, 41:617–631, 1973.

[10] F. Gul and E. Stacchetti. Walrasian Equilibrium with Gross Substitutes. *Journal of Economic Theory*, 87(1):95–124, July 1999.

[11] F. Gul and E. Stacchetti. The English Auction with Differentiated Commodities. *Journal of Economic Theory*, 92(1):66–95, May 2000.

[12] M. O. Jackson. Mechanism Theory. In Ulrich Derigs, editor, *Encyclopedia of Life Support Systems*. EOLSS Publishers: Oxford UK, 2003.

[13] A. S. Kelso and V. P. Crawford. Job Matching, Coalition Formation, and Gross Substitutes. *Econometrica*, 50(6):1483–1504, November 1982.

[14] L. G. Khachiyan. A Polynomial Algorithm in Linear Programming (In Russian). *Doklady Akademii Nauk SSSR; English Translation: Soviet Mathematics Doklady*, 20:191–194, 1979.

[15] A. Kothari, T. Sandholm, and S. Suri. Solving Combinatorial Exchanges: Optimality via a Few Partial Bids. In *AAAI-02 Workshop on AI for Intelligent Business*. Edmonton, 2002.

[16] H. B. Leonard. Elicitation of Honest Preferences for the Assignment of Individuals to Positions. *Journal of Political Economy*, 91(3):461–479, June 1983.

[17] Müller, R. and Schulz, A. S. Combinatorial Auctions. http://web.mit.edu/schulz/www.talks.html, March 2002.

[18] K. Murota and A. Tamura. Application of M-Convex Submodular Flow Problem to Mathematical Economics. *Japan Journal of Industrial and Applied Mathematics*, 20:257–277, 2003.

[19] N. Nissan and I. Segal. The Communication Requirements of Efficient Allocations and Supporting Lindahl Prices. Working Paper, Stanford University, Forthcoming in Journal of Economic Theory, 2005.

[20] D. C. Parkes. Iterative Combinatorial Auctions: Theory and Practice. *Ph. D. Thesis, University of Pennsylvania*, 2001.

[21] M. H. Rothkopf, A. Pekec, and M. R. Harstad. Computationally Manageable Combinatorial Auctions. *Management Science*, 44(8):1131–1147, 1998.

[22] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner Determination in Combinatorial Auction Generalizations. In *First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 69–76. Bologna, Italy, 2002.

[23] W. Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, 16:8–37, 1961.