An Auction-Based Market Equilbrium Algorithm for the Separable Gross Substitutibility Case

Rahul Garg * Sanjiv Kapoor † Vijay Vazirani ‡

Abstract

In this paper we study the problem of market equilibrium. Firstly we consider the model (Fischer model) where there is a supply of money associated with each buyer and a quantity associated with each item. The *market equilibrium problem* is to compute a price vector which ensures market clearing, i.e. the demand of a good equals its supply, and subject to his endowment, each buyer maximizes his utility. Each item has a utility for a given buyer. We assume that the utility function is an increasing, differentiable, concave function. We show that under the assumptions of gross substitutibility, an auction algorithm can determine approximate market clearing.

Not only does this algorithm extends the class of utility functions for which market equilibrium can be determined polynomially, the auction algorithm is efficient and in this model $O(n)$ faster than the auction algorithm for the case of linear utilities in the Arrow-Debreu model.

We outline an extension of our method to the Arrow-Debreu method.

1 Introduction

In this paper we study algorithms for computing market equilibrium in markets with utility functions satisfying the property of gross-substitutibility. The mathematical modeling of market equilibrium was first proposed in 1891 by Fisher [4] where markets were modeled by linear functions. Independently, Walras (1894) proposed the notion of general equilibirum. Walras proposed that a general equilibrium could be achieved by a price-adjustment process called *tatonnement* [15]. The existence of equilibrium prices in a general setting has been established by Arrow and Debreu [1]. The proof is non-constructive and of considerable importance is an effecient computation process which establishes equilibrium. The importance of designing polynomial time schemes has been highlighted in a computer science context by Papadimitriou [14]. Special cases have been dealt with along with related complexity issues in [6]. However, as discussed in Devanur et al. [7], computationally efficient time algorithms had evaded researchers. A polynomial time algorithm for the specific case of linear functions and when the portfolio of the buyer comprises only money, was proposed in Devanur et al. [7] using a primal-dual mechanism. The mechanism used is similar to Kuhn's methodology for bipartite matching [13]. Improvements and generalization of this methodology led to a solution in the general (Arrow-Debreu) case with linear utilities[11, 9]. A

*grahul@in.ibm.com, IBM India Research Lab., Block-I, IIT Campus, Hauz Khas, New Delhi, INDIA - 110016.

†skapoor@iit.edu,Department of Computer Science, Illinois Institute of Technology, Chicago, IL-60616.

‡vazirani@cc.gatech.edu, Georgia Institute of Technology, Chicago, USA.

different mechanism based on an auction schema was proposed in [10]. An exact algorithm for the linear case using the ellipsoid method has been proposed in [12].

In this paper we consider a generalized class of utility functions for the market model. In the market model, defined first by Fischer, there is a supply of money associated withe each buyer and a quantity associated with each item. Each buyer has an associated utility function. In this paper this function is assumed to be increasing differentiable concave unction, linearly separable (w.r.t. the items) and satisfying gross substitutibility. Further the items are divisible. The market equilibrium problem is to compute a price and feasible assignment of goods such that no buyer is induced to change his assignments and market clearing is achieved, i.e. no quantity of goods are left.

Extending the class of utility functions for the Fischer model is important. The linear model is weak for an exact modeling of utility as most demand functions are required to represent satiation of demands. A step in this direction has been taken in [8] where they design a primal-dual algorithm for the spending constraint model, where a step function defines the rate of utility change. The rate of utility change per unit money spent as a function of the money spent defines the utility function.

Our algorithm is based on the auction schema introduced in [10]. While [7] describe a primal dual schema for the equilibrium problem by identifying a primal and dual process which updates item or goods and prices, their proof is based on a characterization of "tight" sets etc., concepts which are very similar to those used in primal-dual methods for matching and flows in bipartite graphs. However matching has a linear programming formulation and a primal-dual mechanism follows naturally from that formulation. The market equilibrium problem is a non-linear problem and has not yet been modeled by a LP. In the paper [10] a parameterized family of linear programs is defined to characterize the problem. The paper uses this formulation to define conditions under which market equilibrium prices achieves market clearing. These conditions arise naturally from complementary slackness conditions. The paper then defines an auction mechanism which approximates the market equilibrium prices to within a tolerance level defined by a parameter ϵ. This provides an efficient methodology for approximating the market equilibrium prices.

We use the auction methodology to achieve approximate clearing for the class of separable functions which are increasing, concave and satisfy gross-substitutibility. Our algorithm achieves a complexity of $O((E/\epsilon)\log(1/\epsilon)\log((ev_{max})/(e_{min}v_{min}))\log m)$ where $e_{min} = \min_i e_i$ and $e = \sum_{i=1}^{n} e_i$ and v_{max}/v_{min}, the ratio of the largest slope to the least slope. We assume that this ratio is bounded. This algorithm is faster by a factor of $O(n)$ as compared to the auction algorithm for the Arrow-Debreu model([10]) and also extends the class of utility functions for which approximate market clearing can be achieved.

Interestingly, while the algorithm's framework remains as simple as in [10], the proof of correctness and convergence are complicated by the general nature of the utility functions. We are able to, in this paper, resolve the convergence to the equilibrium prices via a monotone change in prices for separable increasing functions satisfying gross-substitution and concavity. Gross-substitution has been shown to be a requirement for the convergence of tattonment processes by economists. The problem of equilibrium in the more general class of increasing concave functions (not separable) which satisfy gross-substitutibility is a more challenging problem.

In Section 2 we define the market model and provide a characaterization of gross substitutible functions. In Section 3 we outline our algorithm and prove correctness and the complexity bounds. Finally, we outline (Section 4) an extension to the Arrow-debreu model.

2 Market Model

Consider a market consisting of a set B of n buyers and a set A of m divisible goods. Buyer i has an amount of money equal to e_i. The amount of good j available in the market is b_j. Buyer i has a utility function, $U_i(X)$. where $X = (x_{i1}, x_{i2} \ldots x_{ik})$ represents the current allocation vector of goods. We assume that U_i is non-negative, strictly increasing, differentiable, and concave in the range $[0, a_j]$. Let v_{ij} represent the first derivative of U_i w.r.t. x_{ij} (which is well defined). Assume that the buyers have no utility for money, however they use their money to purchase the goods.

Given prices $p_1, p_2, \ldots, p_m$ of these m goods, a buyer uses its money to purchase goods that maximize its total utility subject to its budget constraint. Thus a buyer i will choose an allocation $X_i \equiv \{x_{ij} : j \in [1..m]\}$ that solves the following buyer program $B_i(P)$:

$$\text{Maximize :} \sum_{j \in [1..m]} u_{ij}(x_{ij}) \tag{1}$$

$$\text{Subject to:} \sum_{j \in [1..m]} x_{ij} p_j \leq e_i \tag{2}$$

$$\forall j : \; x_{ij} \geq 0 \tag{3}$$

This defines a parameterized family of programs, each program defined for a fixed price vector. Since u_{ij} is concave for all i and j, the theory of duality can be used to give the following necessary and sufficient conditions for optimality for a given price P:

$$\sum_{j \in [1..m]} x_{ij} p_j = e_i \tag{4}$$

$$\forall j : \alpha_i p_j \geq v_{ij}(x_{ij}) \tag{5}$$

$$\forall j : x_{ij} > 0 \Rightarrow \alpha_i p_j = v_{ij}(x_{ij}) \tag{6}$$

$$\alpha_i \geq 0, \forall j : x_{ij} \geq 0 \tag{7}$$

We say that the pair $(X, P) \in R_+^{n \times m} \times R_+^m$ forms a market equilibrium if (a) the vector $X_i \in R_+^n$ solves the problem $B_i(P)$ for user i and (b) there is neither a surplus or a deficiency of any good i.e.,

$$\forall j : \sum_{i=1}^{n} x_{ij} = a_j \tag{8}$$

The prices P are called market clearing prices and the allocation X is called the equilibrium allocation at price P.

The equations (8) and (4) imply that all the goods are sold and all the buyers have exhausted their budget. Equations (5) and (6) imply that (a) that every buyer has the same marginal utility per unit price on the goods it gets and (b) every good that a buyer doesn't get gives less marginal utility per unit price.

2.1 Gross Substitutes

Gross substitutes is a well-studied property that has useful economic interpretations. Goods are said to be gross substitutes for a buyer iff increasing the price of a good does not decrease the buyer's demand for other goods. Similarly, goods in an economy are said to be gross substitutes iff increasing the price of a good does not decrease the total demand of other goods. Clearly, if the goods are gross substitute for every buyer, they are gross substitutes in an economy.

We now give a formal definition of gross substitute in our model. Consider the buyer maximization problem $B_i(P)$ where u_{ij} are the utility functions and e_i is the initial endowment of buyer i. Let $X(P) \subset R_+^m$ be the set of optimal solutions of the program $B_i(P)$. Consider another price vector $P' > P$. Goods are gross substitutes for buyer i if and only if for all $x_i \in X_i(P)$ there exists $x'_i \in X_i(P')$ such that $p_j = p'_j \Rightarrow x_{ij} \leq x'_{ij}$.

Assume that u_{ij} is continuous, concave and differentiable for all i and j. Let $v_{ij}(x) = \frac{d}{dx}u_{ij}(x)$. Since u_{ij} is concave, v_{ij} is a non-increasing function. The following result characterizes the class of separable concave gross substitute utility functions.

Lemma 1 *Goods are gross substitutes for buyer i if and only if for all j, $yv_{ij}(y)$ is a non-decreasing function of the scalar y.*

Proof: Consider an optimal solution $x_i \in X_i(P)$. The dual of the program $B_i(P)$ gives the following necessary and sufficient conditions for the optimality of x_i.

$$\sum_{j=1}^{m} x_{ij}p_j = e_i \tag{9}$$

$$\forall j : x_{ij} > 0 \Rightarrow v_{ij}(x_{ij}) = \alpha_i p_j \tag{10}$$

$$\forall j : \alpha_i p_j \geq v_{ij}(x_{ij}) \tag{11}$$

$$\alpha_i \geq 0, x_{ij} \geq 0$$

Equation (10) gives $x_{ij}p_j = x_{ij}v_{ij}/\alpha_i$. Consider $P' > P$. If $v_{ij}(0) < \alpha_i p'_j$ then set x_{ij} to zero, else choose x'_{ij} such that $v_{ij}(x'_{ij}) = \alpha_i p'_j$. By definition, the solution x'_i satisfies the complementary slackness conditions (10). Since $P' > P$, x_i also satisfies (11). v_{ij} is a non-increasing function. Therefore $p'_j > p_j \Rightarrow x'_{ij} \leq x_{ij}$. Now,

$$\begin{aligned} x'_{ij}p'_j &= x'_{ij}v_{ij}(x'_{ij})/\alpha_i \\ &\leq x_{ij}v_{ij}(x_{ij})/\alpha_i \\ &= x_{ij}p_j \end{aligned}$$

The above equations give

$$\sum_{j=1}^{m} x'_{ij}p'_j \leq \sum_{j=1}^{m} x_{ij}p_j = e_i$$

Note that u_{ij} is concave for all j. Therefore, there is an optimal solution x''_i of the program $B_i(P')$ such that $x''_i \geq x'_i$. From the definition of x'_i if $p_j = p'_j$ then $x'_{ij} = x_{ij}$. Therefore $p_j = p'_j \Rightarrow x''_{ij} \geq x_{ij}$ where x_i is an optimal solution of $B_i(P)$ and x''_i is a corresponding optimal solution of $B_i(P')$.

To prove the converse part, assume that there are scalars y and y' such that $y' < y$ and $y'v_{ij}(y') > yv_{ij}(y)$. Choose a price P and an optimal solution x_i of $B_i(P)$ such that $x_{ij} = y$ for some j. Let α_i be the optimal dual solution of $B_i(P)$. Construct a corresponding P' and x'_i such that $x'_{ik} = x_{ik}$, $p'_k = p_k$ for all $k \neq j$, $x'_{ij} = y'$ and $p'_j = p_j v_{ij}(x'_{ij})/v_{ij}(x_{ij})$. Now,

$$\begin{aligned} x'_{ij}p'_j &= x'_{ij}p_j v_{ij}(x'_{ij})/v_{ij}(x_{ij}) \\ &> p_j x_{ij} v_{ij}(x_{ij})/v_{ij}(x_{ij}) \\ &= x_{ij}p_j \end{aligned}$$

So, the solution x_i' satisfies (11) and (10) for price P', but $\sum_{j=1}^m x_{ij}'p_j' > \sum_{j=1}^m x_{ij}p_j = e_i$. Therefore, the optimal dual solution α_i' of $B_i(P')$ will satisfy $\alpha_i' > \alpha_i$. Therefore, the optimal solution x_i'' of $B_i(P)$ will have $x_{ik}'' < x_{ij}'$ for all $k \neq j$, such that $x_{ik} > 0$. Hence the goods will not be gross substitute for buyer i. □

3 An Auction Algorithm for Market Clearing

We now present an ascending price algorithm for discovering the market clearing prices approximately. The algorithm starts with a low price and an initial allocation x_i for all buyers i, such that all the goods are completely allocated and optimal allocation of buyers dominate their current allocation. Now the prices of *overdemanded* items are raised slowly and the current allocation is recomputed, until no item is overdemanded. This approach has a similarity with the Hungarian method of Kuhn [13] for the assignment problem. Unlike the Hungarian method which raises the price of all the goods in a *minimal overdemanded set* by a specific amount, our algorithm raises the price of one good at a time by a fixed multiplicative factor $(1+\epsilon)$, where $\epsilon > 0$ is a small quantity suitably chosen at the beginning of the algorithm. This algorithm has an auction interpretation, where traders outbid each other to acquire goods of their choice by submitting a bid that is a factor $(1+\epsilon)$ of the current winning bid. Prior to this auction algorithms have been proposed for maximum weight matching in bipartite graphs, network flow problems and market clearing with linear utilities [5, 3, 2, 10].

In the intialize procedure (Figure 1) all the items are allocated to the first buyer. Prices are initialized such that (a) the buyer's money is exhausted and (b) the buyer's allocation is optimal at the initial prices. An intitial assignment of dual variables α_i is also required. Instead of α_i we maintain α_{ij}, a separate dual variable for each item, for computational ease. Finally we will relate α_i to α_{ij}.

It is easy to verify that the prices set in procedure `initialize` exhausts the budget of the first buyer. Since $v_{1j}(a_j)$ is assumed to be strictly positive[1], $p_j > 0$ for all j. Therefore the initial value of α_{ij} is well-defined for all i and j.

Also note that $v_{1j}(x_{ij})/p_j = \alpha_{1j} = \alpha_1$ for all j. Hence the initial allocation maximizes the utility of the first buyer.

Define surplus of a buyer i as $r_i = \sum_{j=1}^m (h_{ij}p_j + y_{ij}p_j/(1+\epsilon))$. Define the total surplus in the system as $r = \sum_{i=1}^n r_i$.

The auction algorithm `main` (Figure 1) begins with a buyer i having significant surplus (more than ϵe_i) who tries to acquire items, with utility per unit price more than the current utility per unit price. It outbids other buyers by acquiring items at a higher prices. It raises the prices by a factor $(1+\epsilon)$ if needed. This process continues till total surplus in the economy becomes sufficiently small.

The algorithm maintains the invariants (`I1`) items are fully sold, (`I2`) buyers do not exceed their budget, (`I3, I4`) after completely exhausting its surplus a buyer's utility is close to its optimal utility at the current prices, (`I5`) prices do not fall and (`I6`) total surplus money in the economy does not increase. Figure 2 lists these invariants formally.

[1] This is not a necessary assumption. It is made for simplicity of the presentation. A weaker assumption would be that every item j has a buyer i such that $v_{ij}(a_j) > 0$. The initial allocation may still be found that satisfies the desired properties.

It is easy to check that all the invariants (`I1` through `I6`) are satisfied after the initialization (i.e. after procedure `initialize` has been called.

x_{ij} is modified only in procedure `outbid`. However, the modifications leave the sum $\sum_i x_{ij}$ unchanged. Therefore the invariant `I1` is satisfied throughout the algorithm.

For invariant `I2`, it is sufficient to show that $r_i \geq 0$ for all i. r_i is reduced only in procedure `outbid`. In this procedure, the variable t_2 is chosen such that r_i does not become negative and hence `I2` remains satisfied. Hence the invariant `I2` is satisfied.

For invariant `I6`, note that the only steps that change r are in procedure `outbid`. In these steps, r is reduced by ϵt. Hence `I6` is satisfied in the algorithm. The invariant `I5` is trivially satisfied

We now show that invariants `I3` and `I4` are satisfied by the algorithm.

Lemma 2 *During the approximate auction algorithm the invariants* `I3` *and* `I4` *are always satisfied.*

Proof: The invariants are true initially for all the buyers. We first show invariant `I3`. Note that when α_{ij} is modified in algorithm `main` after calling `outbid`, the invariant is satisfied. Since p_j never decreases, the invariant remains satisfied whenever p_j changes. When x_{ij} is reduced, $v_{ij}(x_{ij})$ increases causing a potential violation of the invariant. In this case, the inner `while` loop of the algorithm will be executed. We argue that when the inner loop ends $\alpha_{ij}p_j \geq v_{ij}(x_{ij})$ for all i, j.

To prove this, consider the time instant z when good j was acquired by buyer i at price p_j. Let a be the quantity of good j acquired by buyer i. Now, $\alpha_{ij}p_j = v_{ij}(a)$. Assume that the amount of good j currently acquired by buyer i is $b < a$. Let the current price of j be p'_j. Choose c such that $v_{ij}(c)/p'_j = v_{ij}(a)/p_j$. It is always possible to do so since $\alpha_{ij}p'_j < v_{ij}(b)$ and $\alpha_{ij}p_j = v_{ij}(a)$. Since $p'_j > p_j$, $c < a$. Now, $cp'_j = cp_j v_{ij}(c)/v_{ij}(a)$. From the assumption that goods are gross substitutes and using Lemma 1 we have $cv_{ij}(c) \leq av_{ij}(a)$. Therefore $cp'_j \leq ap_j$. Therefore the amount of money needed to be spent on j to ensure $v_{ij}(c)p'_j = \alpha_{ij}$ is $p'_j c$ which is no more than $p_j a$; the amount of money spent on j before x_{ij} was reduced. Hence, when the inner loop ends $\alpha_{ij}p'_j \geq v_{ij}(x_{ij})$ for all i and j.

The invariant `I4` is satisfied after initialization. Whenever α_{ij} is changed in `main` $v_{ij}(x_{ij}) = \alpha_{ij}p_j$. Therefore, if x_{ij} is reduced, `I4` remains satisfied. x_{ij} may be increased by a call to `outbid` in the inner loop. However, the parameter α_{ij} ensures that $\alpha_{ij}p_j \leq v_{ij}(x_{ij})$. Moreover, if $x_{ij} > 0$ at the exit of the inner loop, then $\alpha_{ij}p_j = v_{ij}(x_{ij})$. So, if p_j is raised by a factor $(1+\epsilon)$, `I4` will still be satisfied. If p_j rises by more than the factor $1+\epsilon$, x_{ij} will be set to zero and when x_{ij} is increased again $\alpha_{ij}p_j = v_{ij}(x_{ij})$. So, `I4` will remain satisfied in the algorithm.

□

The algorithm ends when the surplus with each buyer is small, i.e. ϵe_i.

Lemma 3 *When the algorithm terminates, approximate optimality is achieved for each buyer, i.e. the allocations and prices give an approximate optimal solution to* $B_i(P)$.

Proof: Consider the time/iteration t after which buyer i has remainder surplus $\leq \epsilon e_i$ till the end of the algorithm. Suppose the optimal allocation of item j is x'_{ij} and the allocation at t is x_{ij}. Since the surplus is bounded

$$u_{ij}(x'_{ij}) - u_{ij}(x_{ij}) = \Delta_{ij} \leq \epsilon e_{ij} v_{ij}(x_{ij})/p_j \leq \epsilon e_{ij} \alpha_{ij}$$

```
procedure initialize
        ∀i, ∀j : h_ij = 0
        ∀i ≠ 1, ∀j : y_ij = 0
        ∀j : y_1j = a_j
        ∀j : α_1j = (Σ_j a_j v_1j(a_j))/e_i
        ∀j : p_j = v_1j(a_j)/α_1
        ∀i ≠ 1 : α_i = v_ij(0)/p_j;  r_i = e_i
        ∀i ≠ 1, ∀j : α_ij = v_ij(x_ij)/p_j
        r_1 = 0
end procedure initialize

algorithm main
        initialize
        while ∃i : r_i > εe_i
              while (r_i > 0) and (∃j : α_ij p_j < v_ij(x_ij))
                    if ∃k : y_kj > 0 then outbid(i, k, j, α_ij)
                    else raise_price(j)
              end while
              j = arg max_l α_il
              if ∃k : y_kj > 0
                    outbid(i, k, j, α_ij/(1 + ε))
                    α_ij = v_ij(x_ij)/p_j
              else raise_price(j)
        end while
end algorithm main

procedure raise_price(j)
        ∀i : y_ij = h_ij; h_ij = 0;
        p_j = (1 + ε)p_j
end procedure raise_price

procedure outbid(i, k, j, α)
        t_1 = y_kj
        t_2 = r_i/p_j
        if (v_ij(a_j) ≥ αp_j) then t_3 = a_j
        else t_3 = min δ : v_ij(x_ij + δ) = αp_j
        t = min(t_1, t_2, t_3)
        h_ij = h_ij + t
        r_i = r_i − tp_j
        y_kj = y_kj − t
        r_k = r_k + tp_j/(1 + ε)
end procedure outbid
```

Figure 1: The auction algorithm

I1:	forall j:	$\sum_i x_{ij} = a_j$
I2:	forall i:	$\sum_j x_{ij} p_j \le e_i$
I3:	forall i:	$r_i = 0 \Rightarrow \alpha_{ij} p_j \ge v_{ij}(x_{ij})$
I4:	forall i, j:	$x_{ij} > 0 \Rightarrow (1+\epsilon) v_{ij}(x_{ij}) \ge \alpha_{ij} p_j$
I5:	forall j:	p_j does not fall
I6:		r does not increase

Figure 2: The invariants in the auction algorithm

where e_{ij} is the portion of the surplus allocated to item j. Summing over all items we get that $\Sigma_j \Delta_{ij} \le \epsilon e_i \alpha_i$, where $\alpha_i = \max_j \alpha_{ij}$. Note that $\alpha_{ij} \ge \alpha_i/(1+\epsilon), \forall j$. α_i for the current prices defines a feasible dual solution. □

3.1 Covergence of the algorithm

The algorithm proceeds in rounds, where in each round each buyer attempts to reduce his surplus.

Lemma 4 *In every round the total unspent money $r = \sum_{i=1}^n r_i$ decreases by a factor of $(1+\epsilon)$.*

Proof:

The value of r is decreased in procedure `outbid` by $t\epsilon p_j$. Buyer i bids until $r_i = 0$. WLOG assume that these bids are of amounts $t_1, t_2, \ldots, t_k$ on items $1, 2, \ldots, k$ at prices $p_1, p_2, \ldots, p_k$. Now we have:

$$\sum_{l=1}^{k} t_l (1+\epsilon) p_l = r_i$$

Reduction in r is given by

$$\Delta r = \sum_{l=1}^{k} t_l \epsilon p_l = \frac{\epsilon}{1+\epsilon} r_i$$

Bidding by buyer i can only increase r_k for another buyer k. Therefore the total reduction in r in one round is given by:

$$\Delta r \ge \sum_{i=1}^{n} \frac{\epsilon}{1+\epsilon} r_i = \frac{\epsilon}{1+\epsilon} r$$

The new value of unspent money r' after every round is related to its old value r as: $r' = \frac{r}{1+\epsilon}$ □

Let $e_{min} = \min_i e_i$ and $e = \sum_{i=1}^n e_i$. If $r < \epsilon e_{min}$ then no buyer has significant money left. Therefore the algorithm is guaranteed to terminate in k rounds where $k = \log(\frac{e}{e_{min}}) \log(\frac{1}{\epsilon}) / \log(1+\epsilon)$.

The price of any item is bounded by:

$$\frac{v_{1j}(a_j)}{\sum_{k=1}^{m} v_{1k}(a_k)} \frac{e_1}{a_j} \leq p_j \leq \frac{e}{a_j}$$

At every step, the price is raised by a factor $(1+\epsilon)$, therefore the total number of price raises for any item is bounded by:

$$\frac{1}{\epsilon} \log\left(\frac{ev}{e_{min} v_{min}}\right)$$

where $v = \sum_{j=1}^{m} v_{1j} i(a_k)$, $e_{min} = \min_i e_i$ and $v_{min} = \min_{ij} v_{1k}(a_k)$. We will assume that the ratio of v and $v_m in$ is bounded. If not, i.e. if v_{min} is zero then we can perturb the utility function $u_{ij}(x_{ij})$ by addition of the term ϵx_{ij} such that the derivative is at least ϵ. Furthermore, the derivative is bounded above by $v_{ik}(a_k)$.

Thus:

Theorem 1 *The auction algorithm terminates in* $O((E/\epsilon) \log(1/\epsilon) \log((ev)/(e_{min} v_{min})) \log m)$ *steps.*

where E is the number of non-zero utilities.

4 Conclusions

The auction mechanism above can be extended to the market model proposed by Arrow and Debreu for the class of utilitiy functions we have considered in this paper. The endowment of each buyer is an initial allocation of items. The prices are intitalized to 1 for each item. Starting with an initial surplus the buyers bid for goods, rasing prices when unable to acquire a particular good. Details are similar to that in [10].

It would be of interest to extend the class of utility functions for which the market equilibrium problem is solvable via the primal-dual auction method.

References

[1] K. Arrow and G. Debreu. Existence of an Equilibrium for a Competitive Economy. *Econometrica*, 22:265–290, 1954.

[2] Vipul Bansal and Rahul Garg. Simultaneous Independent Online Auctions with Discrete Bid Increments. *Electronic Commerce Research Journal: Special issue on Dynamic Pricing Policies in Electronic Commerce*, To Appear.

[3] Dimitri P. Bertsekas. Auction Algorithms for Network Flow Problems: A Tutorial Introduction. *Computational Optimization and Applications*, 1:7–66, 1992.

[4] W. C. Brainard and H. E. Scarf. How to Compute Equilibrium Prices in 1891. Cowles Foundation Discussion Paper (1272), 2000.

[5] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item Auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[6] X. Deng, C. Papadimitriou, and S. Safra. On the Complexity of Equilibria. In *34th ACM Symposium on Theory of Computing (STOC 2002)*, Montreal, Quebec, Canada, May 2002.

[7] N. Devanur, C. Papadimitriou, A. Saberi, and V. Vazirani. Market Equilibrium via a Primal-Dual-Type Algorithm. In *43rd Symposium on Foundations of Computer Science (FOCS 2002)*, pages 389–395, November 2002.

[8] N. Devanur and V. Vazirani. The spending constraint model for market equilibrium: Algorithmic, existence and uniqueness results. In *ACM STOC (to appear)*, 2004.

[9] N. R. Devanur and V. Vazirani. An Improved Approximation Scheme for Computing the Arrow-Debreu Prices for the Linear Case. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2003)*, 2003.

[10] Rahul Garg and Sanjiv Kapoor. Auction algorithms for market equilibrium. In *ACM Symposium on Theory of Computing (STOC)*, 2004.

[11] K. Jain, M. Mahdian, and A. Saberi. Approximating Market Equilibrium. In *Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2003)*, 2003.

[12] Kamal Jain. A Polynomial Time Algorithm for Computing the Arrow-Debreau Market equilibrium for Linear Utilities. Preprint.

[13] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[14] Christos H. Papadimitriou. On the Complexity of the Parity-argument and other Inefficient Proofs of Existence. *Journal of Computer and System Sciences*, 48(3):498–532, June 1994.

[15] L. Walras. Elements of Pure Economics, or the Theory of Social Wealth (in French). Lausanne, Paris, 1874.