Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 378 (2007) 143–152
Theoretical Computer Science
www.elsevier.com/locate/tcs

A primal-dual algorithm for computing Fisher equilibrium in the absence of gross substitutability property[☆]

Dinesh Garg[a,1], Kamal Jain[b,*], Kunal Talwar[b], Vijay V. Vazirani[c]

[a] *Department of Computer Science and Automation, Indian Institute of Science, Bangalore - 560 012, India*
[b] *One Microsoft Way, Redmond, WA 98052, United States*
[c] *College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, United States*

Abstract

We provide the *first strongly polynomial time exact combinatorial algorithm* to compute Fisher equilibrium for the case when utility functions do not satisfy the Gross substitutability property. The motivation for this comes from the work of Kelly, Maulloo, and Tan [F.P. Kelly, A.K. Maulloo, D.K.H. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, Journal of Operational Research (1998)] and Kelly and Vazirani [F.P. Kelly, Vijay V. Vazirani, Rate control as a market equilibrium (2003) (in preparation)] on rate control in communication networks. We consider a tree like network in which root is the source and all the leaf nodes are the sinks. Each sink has got a fixed amount of money which it can use to buy the capacities of the edges in the network. The edges of the network sell their capacities at certain prices. The objective of each edge is to fix a price that can fetch the maximum money for it, and the objective of each sink is to buy capacities on edges in such a way that it can facilitate the sink to pull maximum flow from the source. In this problem, the edges and the sinks play precisely the role of sellers and buyers, respectively, in Fisher's market model. The utility of a buyer (or sink) takes the form of a Leontief function which is known for not satisfying Gross substitutability property. We develop an $O(m^3)$ exact combinatorial algorithm for computing equilibrium prices of the edges. The time taken by our algorithm is independent of the values of sink money and edge capacities. A corollary of our algorithm is that equilibrium prices and flows are rational numbers. Although there are algorithms to solve this problem they are all based on convex programming techniques. To the best of our knowledge, ours is the first strongly polynomial time exact combinatorial algorithm for computing equilibrium prices of Fisher's model under the case when buyers' utility functions do not satisfy gross substitutability property.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Computing market equilibria; Fisher equilibrium; Gross substitutability property; Strongly polynomial time exact algorithm; Combinatorial algorithm; Primal-dual algorithm

☆ A version of this paper has appeared in the First International Workshop on Internet and Network Economics, WINE 2005, Hong Kong, December 2005.

* Corresponding author.
E-mail addresses: dgarg@csa.iisc.ernet.in, dingarg1@in.ibm.com (D. Garg), kamalj@microsoft.com (K. Jain), kunal@microsoft.com (K. Talwar), vazirani@cc.gatech.edu (V.V. Vazirani).
[1] Tel.: +91 80 41775390; fax: +91 80 41922067.

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.02.017

1. Motivation

The primary motivation for our work comes from the work of Kelly [19], Kelly, Maulloo, and Tan [17], and Kelly and Vazirani [18] on rate control in communication networks. The key issue addressed in these papers is how the available bandwidth within the network should be shared among the competing streams of the traffic during the period of congestion. By and large, these papers address the variants of the following problem. Consider a communication network having finite number of users and each user needs to send the data from a source to a sink along a specific path. Each user has a well defined utility function that defines the utility derived by the user if he is able to send the data through the network at a particular rate. Given the facts that communication links of the network have finite capacities and the data traversal paths of the users have links in common, the users cannot be allowed to send the data at arbitrarily high rate because it will cause congestion in the network. Therefore, the issue is to calculate an optimal data transfer rate for each user so that the network can operate without resulting in congestion and at the same, individual users are kept happy in the sense that their utilities get maximized at this optimal rate. Typically, the network does not know the utility functions of the users and, therefore, faces the difficulty in computing these optimal rates. The solution proposed by Kelly et al. [17] suggests to charge the toll from the users for using the links of the networks. That is, assign the prices to the links in such a way that the users find it in their best interest to send the data at optimal rate. In order to realize such a scheme in practice, Kelly et al. [17] proposed two algorithms — primal and dual. In primal scheme, each user incrementally adapts its rate and each link decides its price as a function of total flow it sees. In dual scheme, each link incrementally adapts its price and each user decides its rate by a function of total price he sees.

The above problem can be mapped to the setting of an economic market, where each link can be viewed as a unique good that is put for the sale in the market; the capacity of the link can be viewed as quantity of the good, and the users of the network can be viewed as buyers who are interested in buying different combinations of the goods in different quantities depending on locations of their respective source and sink. Now, computing the prices of the links is equivalent to computing the market equilibrium prices for these goods. This is the major theme behind the investigations made in this paper. In this paper, we address the problem of computing the prices of network resource for congestion avoidance, but from the perspective of market equilibrium. We consider a very special tree link network in which root is the source and all the leaf nodes are the sinks. Each sink has got a fixed amount of money which it can use to buy the capacities of the edges in the network. The edges of the network sell their capacities at certain prices. The objective of each edge is to fix a price that can fetch the maximum money for it, and the objective of each sink is to buy capacities on edges in such a way that it can facilitate the sink to pull maximum flow from the source. In this problem, the edges and the sinks play precisely the role of sellers and buyers, respectively, in Fisher's market model. The utility of a buyer (or sink) takes the form of Leontief function which is known for not satisfying the gross substitutability property. We develop an $O(m^3)$ exact combinatorial algorithm for computing equilibrium prices of the edges. The time taken by our algorithm is independent of the values of sink money and edge capacities. A corollary of our algorithm is that equilibrium prices and flows are rational numbers. To the best of our knowledge, ours is the first strongly polynomial time exact combinatorial algorithm for computing equilibrium prices of Fisher's model under the case when buyers' utility functions do not satisfy the gross substitutability property. However, our algorithm is very much tuned for the special structure of the problem.

2. Outline

The organization of the paper is as follows. In Section 3, we first explain the concept of market equilibrium and then formally define the Fisher equilibrium. Section 4 provides a survey of the literature related to market equilibria computation. In Section 5, we define the problem in a more formal way. In Section 6, we develop an Eisenberg–Gale type convex program and show that the solution of this program constitutes the equilibrium prices of the original problem. Section 7 describes our algorithm and Section 8 presents the analysis of the algorithm. Section 9 constitutes the conclusions and future directions.

3. Market equilibrium

The study of *market equilibrium* has occupied the central stage within both positive (or descriptive) and normative (or prescriptive) economics. The notion of market equilibrium was first proposed by Walras [24] in his pioneering work on the theory of social wealth. This equilibrium is popularly known as *competitive or Walrasian* equilibrium.

Walras formulated the state of the economic system at any point of time as the solution of a system of simultaneous equations representing the demand for goods by consumers, the supply of goods by producers, and the equilibrium condition that supply equals demand on every market. It was assumed that each consumer acts so as to maximize his utility, each producer acts so as to maximize his profit, and perfect competition prevails in the sense that each consumer and producer regards the prices paid and received as independent of his own choices. Walras did not, however, give any conclusive argument to show that the equations, as given, have a solution and thus left behind an open problem of investigating the existence of market equilibrium [1].

3.1. Existence of market equilibrium

The existence of market equilibrium is a deeply investigated problem. In the literature [23,1], this problem has been approached by placing different assumptions on the endowment and utility functions of the agents. An answer to this problem was given in the seminal work of two Nobel Laureates Kenneth Arrow and Gerard Debreu in 1954 [1]. They proved the existence of competitive market equilibria under quite general setting of concave utility functions by applying Kakutani's fixed point theorem. The assumptions made by Arrow–Debreu are much weaker and closer to economic reality than the assumptions made by Wald [23]. The proof outlined by Arrow–Debreu is highly non-constructive in nature and, therefore, the natural question down the line is the existence of an efficient computation process which establishes the equilibrium.

3.2. Fisher equilibrium

Contemporary to Walras, Irving Fisher [22] also independently modelled the market equilibrium in 1891. However, Fisher's model turns out to be the special case of Walras' model. Fisher's market model assumes that there are two kinds of traders in the market: buyers and sellers who trade over a finite set of commodities. Buyers have money and utility functions for goods. Sellers have initial endowment of goods and want to earn money. The equilibrium prices are defined as assignment of prices to goods, so that when every consumer buys an optimal bundle then market clears i.e. all the money is spent and all the goods are sold. If money is also considered as a commodity then it is easy to see that the Fisher model is a special case of the Walras model. For the purpose of defining the Fisher equilibrium in a formal manner, let us consider a market consisting of n buyers and m divisible goods. Buyer i has, initially, a positive amount of money m_i. The amount of good j available in the market is c_j. Let for buyer i, $X_i \subset \Re^m_+$ represents the consumption set, i.e., the set of bundles of m goods which buyer i can consume. Let $u_i : X_i \mapsto \Re$ be the utility function for buyer i. Given the prices $p_1, \ldots, p_m$, it is easy to compute the bundle $x_i \in X_i$ which will maximize buyer i's utility subject to his budget constraint. The prices $p^*_1, \ldots, p^*_m$ are said to be *Fisher equilibrium prices* if after each buyer is assigned such an optimal bundle, there is no surplus or deficiency of any goods. If x^*_{ij} denotes the amount of good j bought by buyer i in his optimal bundle x^*_i at prices $p^*_1, \ldots, p^*_m$, then it can be verified [1] that this price vector $p^*_1, \ldots, p^*_m$ is Fisher equilibrium iff it satisfies the following conditions:

- $\left(\sum_{i=1}^n x^*_{ij} - c_j\right) \leq 0 \; \forall j = 1, \ldots, m$;
- $p^*_j \left(\sum_{i=1}^n x^*_{ij} - c_j\right) = 0 \; \forall j = 1, \ldots, m$
- x^*_i maximizes $u_i(x_i)$ over the set $\left\{x_i \in X_i \mid \sum_{j=1}^m x_{ij} p^*_j \leq m_i\right\}$ for every $i = 1, \ldots, n$
- $p^*_j \geq 0 \; \forall j = 1, \ldots, m$.

The Arrow–Debreu theorem [1] says that if the utility functions $u_i(.)$ are concave then such an equilibrium price vector always exists. It is easy to see that in equilibrium, each buyer must spend his full budget.

3.3. Gross substitutability

Gross substitutability is a well-studied property that has useful economic interpretation [11]. 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. The demand of a buyer is basically the vector of quantity of each good that the buyer would like to buy in order to maximize his/her utility with respect to the given prices and the budget constraint. 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 substitutes for every buyer, they are Gross substitutes in the economy. Note that, whether goods are Gross substitutes or not for a given buyer i depends solely on his own utility function $u_i(x_i)$. It can be shown that not all concave utility functions satisfy this property. Computing Fisher equilibrium when the buyers utility functions do not satisfy Gross substitutability property is a far more difficult problem than the case when they satisfy this property. A frequently arising utility function that does not satisfy this property is a Leontief utility function. A Leontief utility function for buyer i in Fisher's model looks like this: $u_i(x_i) = \min_j(x_{ij})$. We will be using this utility function in our problem.

4. Literature related to computation of market equilibria

The recent papers by Papadimitriou [21] and Deng et al. [5,4] have raised and partly answered the question of efficient computability of Walrasian equilibrium. These papers have sparked the avalanche of research in this area and have initiated altogether a new frontier on algorithmic theory for market equilibria. Much of the literature in this direction is aimed at developing a polynomial time algorithm for computing the Walrasian equilibrium either accurately or approximately under different simplifying assumptions. We prefer to classify the existing literature into three categories.

The first category includes the literature which address the problem of developing efficient algorithm for computing the Fisher equilibrium under the linear utility case. Way back in 1960s, Eisenberg and Gale [8,9] first formulated the problem of Fisher equilibrium with linear utilities as a convex optimization problem. They constructed an aggregated concave objective function that is maximized at the equilibrium. Thus, finding an equilibrium in this setting becomes solving a convex optimization problem. This implicitly implies a polynomial time (approximation) algorithm, in the sense of numerical computing, i.e., polynomial time in input size and $\log(1/\epsilon)$, where ϵ is the precision in computation. The first exact polynomial-time algorithm for this setting was proposed by Devanur et al. [6] which uses primal-dual type approach.

The second category includes the literature that address the problem of developing efficient algorithm for computing Walrasian equilibrium when the production firms are absent and the utilities are linear. The first algorithm for this setting was proposed by Deng et al. [5] where they developed a polynomial-time algorithm for this setting with an additional assumption that the number of goods or agents is bounded. In the recent past, a number of approximation algorithms have emerged for this setting. Newman and Primak [20] developed an approximate algorithm which runs the ellipsoid algorithm on an infinite linear program. Recently an FPTAS was proposed for the same setting by Jain, Mahdian, and Saberi [13]. In this FPTAS, the running time depends on the size of the numbers representing the utilities and endowments of the traders. Recently, Devanur and Vazirani [7] developed a strongly polynomial time approximation scheme for the same problem and their algorithm builds upon the main ideas behind the algorithm in [6]. Garg and Kapoor [10] have also proposed a fully polynomial time approximation algorithm for the same problem but they use an auction based approach. Jain [14] have shown that computing the Walrasian equilibrium for this setting is indeed equivalent to solving a convex optimization problem. Jain has used this convex program to develop the first polynomial time exact algorithm for computing the Walrasian equilibrium for this setting. This algorithm uses ellipsoid algorithm and simultaneous diophantine approximation. Thus, the algorithm proposed by Jain [14] settles the open problem of computing the market equilibria efficiently under the scenario when utilities are linear and production firms are absent. However, designing efficient market equilibrium algorithms for general concave utility functions is still an open problem. Using the convex program proposed by Jain [14], Ye [25] developed a practical interior point algorithm to find the Walrasian equilibrium under the same setting.

The third category includes the literature that address the problem of developing efficient algorithm for computing Walrasian equilibrium when the production firms are absent and the utility functions do not satisfy the Gross substitutability (the linear utility functions satisfy the Gross substitutability property). The problem of computing the market equilibrium with linear utilities is comparatively simpler because they satisfy the Gross substitutability, i.e., increasing the price of one good cannot decrease the demand for another. Hence for such utility functions, monotonically raising prices suffices. In contract, concave and even piecewise-linear and concave, utility functions do not satisfy Gross substitutability, hence requiring the more involved process of increasing and decreasing prices. Therefore, designing market equilibrium algorithms for them remains an outstanding open problem. In a recent paper, Ye [26] addresses the problem of computing Fisher equilibrium with piecewise linear utility functions, which include Leontief's utility functions. In this paper, the author shows that the Fisher model again reduces to the general analytic center model discussed in [25], and the same linear programming complexity bound applies to approximating its equilibrium. Another article in the similar spirit is due to Codenotti and Vardarajan [3] where the authors propose a polynomial time approximate algorithm for computation of market equilibrium for the Fisher model where the traders have Leontief utility functions. Their result builds upon the construction of a constrained nonlinear maximization problem. In another related paper, Codenotti, Pemmaraju, and Varadarajan [2] propose the first polynomial-time algorithms for the same setting but with the assumption that traders utility functions satisfy the weak Gross substitutability property.

5. Problem statement

The precise problem we solve is the following: Let $T = (V, E)$ be a tree with integer capacities on edges. Let root of the tree be a source node and $T = \{t_1, \ldots, t_k\}$ be the sink nodes. Without loss of generality, we can assume that each sink t_i is the leaf node and conversely each leaf node is a sink.[2] The sinks have budgets $m_1, \ldots, m_k$, respectively. Each sink can use its budget to buy the capacities of the edges in the network. The edges of the network sell their capacities at certain prices. The objective of each edge is to fix a price which can fetch maximum money for it and, at the same time, the objective of each sink is to buy capacities on edges in such a way that it can facilitate the sink to pull maximum flow from the source. In order to map this problem to Fisher's market model, we view edges as the sellers who are trying to sell their capacities and sinks as buyers who are trying to buy the capacities on the edges and whose Leontief utilities are given by $u_i(x_i) = \min_{j|j\in P_i}(x_{ij})$, where x_{ij} is amount of the capacity bought by sink t_i on edge j and P_i is the collection of edges that forms a unique path from source to sink t_i. The problem is to determine Fisher equilibrium prices for all the edges, and flows from the source to the sinks. It is easy to verify from the definition of Fisher equilibrium given in Section 3.2 that edge prices $p_e \forall e \in E$ and flows f_i from source to each sink t_i, form Fisher equilibrium iff they satisfy the following conditions:

1. $f_e \le c_e \, \forall e \in E$.
2. For any edge $e \in E$, if $p_e > 0$, then $f_e = c_e$.
3. For any edge $e \in E$, if $f_e < c_e$, then $p_e = 0$
4. $f_i = \frac{m_i}{\left(\sum_{e|e\in P_i} p_e\right)}$
5. $p_e \ge 0 \, \forall e \in E,\ f_i \ge 0 \, \forall t_i \in T$

where c_e and f_e are the capacity and total flow, respectively for the edge $e \in E$. The P_i is the set of edges which forms a unique path in the tree T from the source to the sink t_i. For each edge $e \in E$, the total flow f_e is given by flow conservation equation $f_e = \sum_{t_i|e\in P_i} f_i$. Note that the first condition corresponds to capacity constraint, the second, third, and fourth conditions correspond to three equilibrium conditions, and the fifth condition is obviously a nonnegativity constraint. Also note that in view of the first condition, the second condition can be relaxed slightly and $f_e = c_e$ can be replaced by $f_e \ge c_e$. Thus, we can say that our problem is to design an algorithm which will take a tree $T = (V, E)$, the capacities $c_1, \ldots, c_m$ of all the edges, and the budgets $m_1, \ldots, m_k$ of all the sink nodes as the input and will return the numbers $p_e \, \forall e \in E,\ f_e \, \forall e \in E,\ f_i \, \forall t_i \in T$ such that these numbers satisfy the above five conditions.

6. Convex program and equilibrium

It is interesting to see that the problem of computing market equilibrium that we sketched in the previous section is captured by the following Eisenberg–Gale type convex program which maximizes the sum of logarithms of flows, weighted by budgets, subject to capacity constraints on the flows.

$$\begin{aligned} \text{Maximize} \quad & \sum_{t_i\in T} m_i \log f_i \qquad (1)\\ \text{subject to} \quad & \sum_{t_i|e\in P_i} f_i \le c_e \, \forall e \in E \\ & f_i \ge 0 \, \forall t_i \in T. \end{aligned}$$

[2] If a sink t_i is an internal node then we can always add an additional leaf edge of infinity capacity at that particular internal node and push the sink t_i to this newly generated leaf node. Similarly, if a leaf node is not a sink then we can just remove the corresponding leaf edge from the tree.

Let p_e's be the dual variables (also called Lagrange multipliers) corresponding to the first set of constraints; we will interpret these as prices of the edges. Using KKT conditions, one can show that f_i's and p_e's form an optimal solution to the primal problem (1) and its corresponding dual problem, respectively, iff they satisfy the following four conditions.

1. Primal Feasibility
$\sum_{t_i|e\in P_i} f_i \le c_e \; \forall e \in E$
$f_i \ge 0 \; \forall t_i \in T$
2. Dual Feasibility
$p_e \ge 0 \; \forall e \in E$
3. Complementary Slackness
$p_e(c_e - \sum_{t_i|e\in P_i} f_i) = 0 \; \forall e \in E$
4. Lagrange Optimality
$f_i = \frac{m_i}{\left(\sum_{e|e\in P_i} p_e\right)} \; \forall t_i \in T.$

It is easy to verify that the above conditions are precisely the same as the six conditions mentioned in the previous section which must be satisfied by the output of our algorithm. Thus, we basically develop a primal dual type algorithm to solve the above optimization problem where the primal variables are flows and the dual variables are edge prices. Almost all the known primal-dual type algorithms for solving such programs operate by raising dual variables greedily. An important exception is Edmonds' general weighted graph matching algorithm which uses a sophisticated increase–decrease process to find an optimal dual. A possible exception could be [15] where the new auxiliary variables only increase but the original dual variables may increase–decrease. In fact in our algorithm too there are auxiliary variables which only increase. In this respect, our algorithm resembles Edmonds' algorithm, with edge prices increasing and decreasing in order to find equilibrium prices for the edges. In our algorithm, the equilibrium prices are essentially unique in the following sense: for each sink t_i, the cost of the path from the source to the sink is the same in all the equilibria. We denote the cost of the path from the source to the sink t_i as price(t_i) and it is given by $\sum_{e|e\in P_i} p_e$. A corollary of our algorithm is that equilibrium prices and flows are rational numbers. We show that this does not hold even if there are just two sources in the tree.

7. The algorithm

Here is a high level description of our algorithm: We start with zero prices of the edges and iteratively change prices. Since prices are zero initially, all the sinks draw infinity flows and hence all the edges are over-saturated; we iteratively decrease the number of such edges. We maintain the following invariants throughout our algorithm

I1. For any edge $e \in E$, if $p_e > 0$, then $f_e \ge c_e$.
I2. For any edge $e \in E$, if $f_e < c_e$, then $p_e = 0$
I3. $f_i = \frac{m_i}{\left(\sum_{e|e\in P_i} p_e\right)} \; \forall t_i \in T$
I4. $p_e \ge 0 \; \forall e \in E,\; f_i \ge 0 \; \forall t_i \in T$
I5. $f_e = \sum_{t_i|e\in P_i} f_i \; \forall e \in E.$

Thus, we maintain dual feasibility, complementary slackness, and Lagrange optimality conditions throughout the algorithm. The only condition that is not maintained is primal feasibility. Thus, our algorithm would terminate at the point where primal feasibility is attained. At such point, all KKT conditions would have been met and the current values of f_i and p_e would be the desired solution. At any instant during the course of the algorithm, each edge would be marked either red or green. If an edge e is marked as green then it means that primal feasibility condition $f_e \le c_e$ is being satisfied at that edge. Initially, all the edges are red and the subroutine **make-green** converts at least one red edge into green. The Algorithm 1 below gives the high level pseudo code for the algorithm. Later we will show that this algorithm also maintains the following invariants.

I6. The parameter price(t_i) is non-decreasing throughout the algorithm for each sink t_i, and flows f_i are non-increasing.
I7. At any instant, the price of a red edge is zero and the set of red edges forms a subtree containing the root of the original tree.

In what follows we describe the subroutine **make-green** which is the crux of the algorithm.

Algorithm 1 Market Equilibrium

Procedure *Market_Equilibrium*$(V, E, c_1, \ldots, c_m, m_1, \ldots, m_k)$:

$p_e \leftarrow 0\ \forall e \in E$
$f_i \leftarrow \infty\ \forall t_i \in T$
$mark[e] \leftarrow red\ \forall e \in E$
if ($\exists e \in E$ such that $mark[e] = red$) **then**
    make-green$(V, E, c, m, mark[.], f, p)$
else
    Stop
end if

7.1. Subroutine ***make-green***

Before we can describe this subroutine, we need to define two concepts.

Definition 1 (*Lower Vertex*). We will say that vertex u is *lower than* v if vertex u is a descendent of vertex v in the tree T. Similarly edge (u, v) is *lower than* (a, b) if vertex u is a descendent of vertex b, vertex v is a children of vertex u, and vertex b is a children of vertex a.

Definition 2 (*Feasible Flow*). Consider a partition $\mathcal{P}$ of E into *red* and *green* edges. Let p_e be the current assignment of prices to the edges. Let this partition-price pair $(\mathcal{P}, p)$ satisfies the invariant **I7**. A flow f is said to be *feasible* for this partition-price pair $(\mathcal{P}, p)$ if it satisfies the five invariant conditions **I1** through **I5** mentioned earlier.

In view of the above definitions, we first explain the subroutine **make-green** at a very high level in terms of its input parameters and output parameters. Apart from others, the following are the critical input parameters to each call to the subroutine **make-green**

1. A partition-price pair $(\mathcal{P}, p)$ which satisfies the invariant **I7**.
2. A feasible flow, f, for this partition-price pair.

The subroutine **make-green** converts at least one red edge into green and produces the following parameters as the output

1. A new partition-price pair $(\mathcal{P}', p')$.
2. A feasible flow, f', for the new partition-price pair $(\mathcal{P}', p')$.

The partition-price pair $(\mathcal{P}', p')$ returned by **make-green** respects the invariant **I7**. Later we will show in Lemma 4 that each call to the subroutine **make-green** requires $O(m^2)$ computations, where m is the number of edges in the tree. The subroutine **make-green** accomplishes this via an involved process that increases and decreases prices on edges. By looking at Algorithm 1, one can easily verify that a total of $O(m)$ calls of **make-green** suffice and this leads to the following results.

Theorem 1. *$O(m^3)$ computations suffice to find an optimal solution to convex program* (1) *and the corresponding equilibrium prices and flows for trees. Furthermore, such trees always admit a rational solution.*

Theorem 2. *The time taken by the Algorithm* 1 *does not depend on actual values of sink money and edge capacities and therefore the Algorithm* 1 *is strongly polynomial.*

Now, we give a detailed explanation of how the subroutine **make-green** works. The subroutine **make-green** works iteratively. Every time the subroutine **make-green** is called, it starts by picking a topologically lowest red edge (u, v), i.e., all edges lower than (u, v) are green. If in the given feasible flow f, the edge (u, v) carries a flow less than or equal to its capacity then turn this edge into green and return. This will give a new partition with an extra green edge, the new prices for edges, and a feasible flow for the new partition-price pair.

Otherwise, let Z be the set of vertices reachable from v by the following green zero price edges. Z is called a *zero component* of the edge (u, v). Let A be the set of edges both whose end points are in Z. Let B be the set of edges incident on Z, excluding the edge (u, v). Clearly, all edges in A must be green and must have zero prices and all edges in B must be green and must have positive prices. Let $T_1 \subset T$ be the set of sinks that are sitting inside the zero component Z, $T_2 \subset T$ be the set of sinks that are sitting outside the zero component Z but are lower than vertex v, and $T_3 \subset T$ be the set of remaining sinks. Note that it is quite possible that T_1 is an empty set.

Now the idea is following. Increase the price of the red edge (u, v) and decrease the price of each edge in the set B by the same amount. This will result in price(t_i) to be undisturbed for all the sinks in the set T_2 as well as in the set T_3. However, this would increase the price(t_i) for all the sinks in the set T_1. After changing these prices, we recompute the flows f_i for each sink by using the formula given by invariant **I3**. Note that f_i would remain unchanged for all the sinks in the set T_2 as well as in the set T_3. However, f_i would decrease for all the sinks in the set T_1. Thus, the flow on edge (u, v) may decrease.

The obvious question now is how much to increase the price of red edge (u, v). Note that during this process of increasing the price of edge (u, v), any one of the following two events may occur.

E1: The flow on edge (u, v) hits its capacity.
E2: The price of some edge in the set B goes to zero.

If the event **E1** occurs then turn the red edge (u, v) into green and this will give a new partition with an extra green edge, the new prices for edges, and a feasible flow for the new partition-price pair. If the event **E2** occurs then we freeze the price of edge (u, v) and the edges in the set B at their current values. Now we need to reconstruct the zero component Z and the sets A, B, T_1, T_2, and T_3 because the edge whose price became zero as a consequence of E2 needs to be added to B. We now repeat the process of increasing the price of red edge (u, v) from its current value. In order to identify which one of these two events has occurred, we maintain the following quantities:

$$m_{in} = \sum_{t_i \in T_1} m_i, \qquad f_{green} = \sum_{t_i \in T_2} f_i, \qquad p_{min} = \min(p_e | e \in B).$$

Now we distinguish between two cases:

- **Case 1:** $(f_{green} > c_{(u,v)})$
  It is easy to see that if this is the case then the next event would be **E2**.
- **Case 2:** $(f_{green} \le c_{(u,v)})$
  Under this case, it is easy to verify that if $\frac{m_{in}}{C_{(u,v)} - f_{green}} - p_{(u,v)} \le p_{min}$ then event **E1** will occur, otherwise event **E2** will occur.

The Algorithm 2 gives the pseudo code for the subroutine **make-green** .

8. Analysis

In this section, we state key observations and facts pertaining to the algorithm. We have omitted the proofs because they are quite straightforward and follow directly from the way we have structured the algorithm.

Lemma 1. *If an edge e becomes green in an iteration of Algorithm* 1 *then it will remain green in all the future iterations of the Algorithm* 1.

Lemma 2. *The invariant* **I7** *is maintained throughout the Algorithm* 1.

Lemma 3. *The invariant* **I6** *is maintained throughout the Algorithm* 1.

Theorem 3. *The prices p and flow f attained by the Algorithm* 1 *at its termination are equilibrium prices and flow.*

Lemma 4. *The running time of each call of the subroutine* **make-green** *is bounded by $O(m^2)$ and also this running time is independent of actual values of sink money and edge capacities.*

8.1. Rational solution

The results in this section highlights the fact that the solution given by our algorithm consists of only rational numbers.

Theorem 4. *As long as all the sinks in the tree draw their flows from single source, the convex program* (1) *has a rational solution.*

Algorithm 2 Subroutine make-green

Procedure $make-green(V, E, m, mark[.], f, p)$:

1: find topologically lowest edge e such that $mark[e] = red$
2: **if** $f_e \leq c_e$ **then**
3: $mark[e] \leftarrow green$
4: return
5: **else**
6: construct zero component Z, sets B, T_1, and T_2 for the edge e
7: $m_{in} \leftarrow \sum_{t_i \in T_1} m_i$
8: $f_{green} \leftarrow \sum_{t_i \in T_2} f_i$
9: $p_{min} \leftarrow \min(p_e | e \in B)$
10: **if** $f_{green} > c_e$ **then**
11: $p_{e'} \leftarrow p_{e'} - p_{min} \; \forall e' \in B$
12: $p_e \leftarrow p_e + p_{min}$
13: $f_{t_i} \leftarrow \frac{m_i}{p_e} \; \forall t_i \in T_1$
14: Go to line number 6
15: **else if** $\left(\frac{m_{in}}{C_e - f_{green}} - p_e\right) \leq p_{min}$ **then**
16: $p_{e'} \leftarrow p_{e'} - \left(\frac{m_{in}}{C_e - f_{green}} - p_e\right) \; \forall e' \in B$
17: $p_e \leftarrow \frac{m_{in}}{C_e - f_{green}}$
18: $f_{t_i} \leftarrow \frac{m_i}{p_e} \; \forall t_i \in T_1$
19: $mark[e] \leftarrow green$
20: return
21: **else**
22: Go to line number 12
23: **end if**
24: **end if**

Proof. This follows from Theorem 3 and Lemma 4.

Lemma 5. *Even if there are multiple equilibria for convex program* (1)*, the path price for each sink t_i, i.e. $price(t_i)$, is unique.*

Proof. Note that the objective function of the convex program (1) is strictly concave. This implies that this program must have a unique optimal solution, say f^*. However, there may be multiple optimal solutions of the corresponding dual program. Now by virtue of *Lagrange Optimality* condition for this convex program, we can claim that even if there are multiple equilibria (f^*, p^*) for the convex program (1), the path price for each sink t_i, i.e. $price(t_i)$, is unique.

Lemma 6. *The path price for each sink t_i, i.e. $price(t_i)$, is a rational number.*

Proof. This follows from Theorem 3 and Lemma 4.

8.2. Multiple sources and irrational solution

If there are multiple sources present in the tree then it may give rise to irrational equilibrium prices and flows. For example, consider a tree on three nodes, $\{a, b, c\}$ and two edges $\{ab, bc\}$. Let the capacity of (a, b) be one unit and the capacity of (b, c) be two units. The source sink pairs together with their budgets are: $(a, b, 1)$, $(a, c, 1)$, $(b, c, 1)$. Then the equilibrium price for ab is $\sqrt{3}$ and for bc it is $\frac{\sqrt{3}}{1+\sqrt{3}}$.

9. Discussion

The primal-dual schema has been very successful in obtaining exact and approximation algorithms for solving linear programs arising from combinatorial optimization problems. Ref. [6] and our paper seem to indicate that it is worthwhile applying this schema to solving specific classes of nonlinear programs. There are several interesting convex programs in the Eisenberg–Gale family itself, see [16] and the references therein. Another family of nonlinear programs deserving immediate attention is semidefinite programs. Considering the large running time required to solve such programs, it will be very nice to derive a combinatorial approximation algorithm for MAX CUT for instance, achieving the same approximation factor as [12].

Extending our algorithm to handling arbitrary directed cyclic graphs is another challenging open problem. Also interesting will be to obtain approximation algorithms for the cases where the solution is irrational. Another interesting question is to obtain an auction-based algorithm for tree (or acyclic graphs) along the lines of [10]. Such an algorithm will be more useful in practice than our current algorithm.

Acknowledgment

We would like to thank Philip A. Chou for several useful discussions.

References

[1] K. Arrow, G. Debreu, Existence of an equilibrium for a competitive economy, Econometrica 22 (1954) 265–290.
[2] B. Codenotti, S. Pemmaraju, K. Varadarajan, On the polynomial time computation of equilibria for certain exchange economies, in: Proceedings of SODA, 2004.
[3] B. Codenotti, K. Varadarajan, Efficient computation of equilibrium prices for markets with leontief utilities, in: Proceedings of ICALP, 2004.
[4] X. Deng, C.H. Papadimitriou, S. Safra, On the complexity of price equilibria, Journal of Computer and System Sciences 67 (2003) 311–324.
[5] X. Deng, C.H. Papadimitriou, S. Safra, On the complexity of equilibria, in: STOC 2002, 2002.
[6] Nikhil R. Devanur, Christos H. Papadimitriou, Amin Saberi, Vijay V. Vazirani, Market equilibrium via a primal-dual-type algorithm, in: 43rd Symposium on Foundations of Computer Science, FOCS 2002, November 2002, pp. 389–395.
[7] Nikhil R. Devanur, Vijay V. Vazirani, An improved approximation scheme for the computing Arrow–Debreu prices in the linear case, in: Proceedings of Foundations of Software Technology and Theoretical Computer Science, 2003, 2002.
[8] E. Eisenberg, D. Gale, Consensus of subjective probabilities: The Pari–Mutuel method, Annals of Mathematical Statistics 30 (1959) 165–168.
[9] D. Gale, Theory of Linear Economic Models, McGraw-Hill, New York, 1960.
[10] Rahul Garg, Sanjiv Kapoor, Auction algorithms for market equilibrium, in: Proceedings of the 36th Annual ACM Symposium on the Theory of Computing, 2004.
[11] Rahul Garg, Sanjiv Kapoor, Vijay Vazirani, An auction-based market equilibrium algorithms for the separable gross substitutability case, in: APPROX-RANDOM 2004, 2004, pp. 128–138.
[12] M.X. Goemans, D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, Journal of ACM 42 (1995) 1115–1145.
[13] K. Jain, M. Mahdian, A. Saberi, Approximating market equilibrium, in: Workshop on Approximation Algorithms for Combinatorial Optimization, APPROX 2003, 2003.
[14] Kamal Jain, A polynomial time algorithm for computing the Arrow–Debreu market equilibrium for linear utilities, in: FOCS, 2004.
[15] Kamal Jain, Mohammad Taghi Hajiaghayi, The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema, in: Manuscript, 2004.
[16] Kamal Jain, V.V. Vazirani, Yinyu Ye, Market equilibria for homothetic, quasi-concave utilities and economies of scale in production, in: Proceedings, 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005.
[17] F.P. Kelly, A.K. Maulloo, D.K.H. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, Journal of Operational Research (1998).
[18] F.P. Kelly, Vijay V. Vazirani, Rate control as a market equilibrium (2003) (in preparation).
[19] Frank Kelly, Charging and rate control for elastic traffic, European Transactions on Telecom 8 (1997) 33–37.
[20] D.J. Newman, M.E. Primak, Complexity of circumscribed and inscribed ellipsoid methods for solving equilibrium models, Applied Mathematics and Computation 52 (1992) 223–231.
[21] Christos H. Papadimitriou, Algorithms, games, and the internet, in: ACM STOC'01, Hersonissos, Crete, Greece, 6–8 July 2001.
[22] H. Scarf, The Computation of Economic Equilibria (With Collaboration of T. Hansen), in: Cowles Foundation Monograph, no. 24, Yale University Press, New Haven, 1973.
[23] A. Wald, Über einige gleichungssysteme der mathematischen Ökonomie, Zeitschrift fur Nationalokonomie 7 (1936) 637–670. Translated as On Some Systems of Equations of Mathematical Economics, Econometrica 19 (1951) 368–403.
[24] L. Walras, Éléments d'économie politique pure ou théorie de la richesse sociale (Elements of Pure Economics, or The Theory of Social Wealth), Lausanne, Paris, 1874 (1899, 4th Ed., 1926, Rev. Ed., 1954, Engl. Transl.).
[25] Yinyu Ye, A path to Arrow–Debreu competitive market equilibrium, Working Paper, Department of Management Science and Engineering, Stanford University, Stanford, CA 94305, 2004, Mathematical Programming (2004) (in press).
[26] Yinyu Ye, Exchange market equilibria with leontief's utility: Freedom of pricing leads to rationality, in: Proceedings of 1st Workshop on Internet and Network Economics, WINE, Hong Kong, December 2005.