Games and Economic Behavior 70 (2010) 84–106

Contents lists available at ScienceDirect

Games and Economic Behavior

www.elsevier.com/locate/geb

Eisenberg–Gale markets: Algorithms and game-theoretic properties

Kamal Jain [a], Vijay V. Vazirani [b,*,1]

[a] One Microsoft Way, Redmond, WA 98007, USA

[b] College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, United States

ARTICLE INFO

Article history:
Received 28 November 2007
Available online 24 December 2008

JEL classification:
C62
C63
C68

Keywords:
General equilibrium theory
Fisher market model
Combinatorial algorithm
Primal-dual algorithm
Convex program
Resource allocation
Ascending price auctions
Weak gross substitutability
Competition monotonicity
Fairness

ABSTRACT

We define a new class of markets, the *Eisenberg–Gale markets*. This class contains Fisher's linear market, markets from the resource allocation framework of Kelly [Kelly, F.P., 1997. Charging and rate control for elastic traffic. Europ. Transactions Telecommunications 8, 33–37], as well as numerous interesting new markets. We obtain combinatorial, strongly polynomial algorithms for several markets in this class. Our algorithms have a simple description as ascending price auctions. Our algorithms lead to insights into game-theoretic properties of these markets, such as efficiency, fairness, and competition monotonicity. They also help determine if these markets always have rational equilibria. A classification of Eisenberg–Gale markets w.r.t. these properties reveals a surprisingly rich set of possibilities.

1. Introduction

General equilibrium theory, which answered several crucial questions about the effectiveness of pricing mechanisms within a formal mathematical setting, suffered from one major deficiency—it was an essentially non-algorithmic theory. In view of the fact that numerous new markets have recently emerged on the Internet and massive computational power is available for running these markets in a centralized or distributed manner, the need for reviving the study of market equilibria via an inherently algorithmic approach is well recognized.

A complexity-theoretic study of computing equilibria was initiated by Megiddo (1988) (see also the subsequent paper, Megiddo and Papadimitriou, 1991). Among the first algorithmic results to be obtained within theoretical computer science was a polynomial time algorithm (Devanur et al., 2008) for computing equilibrium allocations and prices for Fisher's market model (Brainard and Scarf, 2000), under the assumption that buyers' utilities are linear functions of their allocations. It turns out that a remarkable nonlinear convex program, due to Eisenberg and Gale (1959), captures, as its optimal solution, equilibrium allocations for this case; the algorithm of Devanur et al. (2008) therefore finds an optimal solution to this program in polynomial time.

* Corresponding author. Fax: +1 404 385 3157.
E-mail addresses: kamalj@microsoft.com (K. Jain), vazirani@cc.gatech.edu (V.V. Vazirani).

1 Supported by NSF Grant CCF-0728640.

0899-8256/$ – see front matter
doi:10.1016/j.geb.2008.11.011

In a different context, that of modeling and understanding TCP congestion control,[2] Kelly (1997) defined a class of resource allocation markets and gave a convex program that captures equilibrium allocations for his model. Interestingly enough, Kelly's program has the same structure as the Eisenberg–Gale program—similar to the latter program, it also maximizes a money-weighted geometric mean of buyers' utilities subject to linear packing constraints.

In a short note, Kelly and Vazirani (2002) observed that a suitable generalization of Kelly's model (see details in Section 16) generalizes Fisher's linear case and asked the open problem of finding polynomial time discrete algorithms for computing equilibria for Kelly's model. The *flow market* is of special significance within this framework. It consists of a network, with link capacities specified, and source-sink pairs of nodes, each with an initial endowment of money; allocations in this market are flows from each source to the corresponding sink. The problem is to find equilibrium flows and prices of edges (in the context of TCP, the latter can be viewed as drop rates at links).

Kelly's model attracted much theoretical study, partly with a view to designing next generation protocols. Continuous time algorithms (though not having polynomial running time), for finding equilibrium flows in the flow market were given by Kelly et al. (1998) (see also Wang et al., 2005, for more recent work along these lines). In asking for discrete algorithms for computing equilibria, Kelly and Vazirani (2002) stated, "Continuous time algorithms similar to TCP are known, but insights from discrete algorithms may be provocative."

Recently, some progress was made on this issue: Garg et al. (2005) gave a strongly polynomial time algorithm for the special case of the flow market when the network is restricted to a tree having one source and multiple sinks. In view of the fairly involved algorithm developed in Garg et al. (2005), they had stated the open problem of extending their result to directed acyclic networks with one source and multiple sinks (Example 1 may help the reader appreciate some of the intricacies of this case).

Our first result in this paper is a strongly polynomial algorithm for computing equilibrium flows and prices of edges for a flow market in any network, directed or undirected, with one source and multiple sinks.

A surprising property of the Eisenberg–Gale program (and of Fisher's linear case) is that despite its nonlinearity, it always has a rational solution if all the parameters in the instance are rational. Our algorithm implies that the same is also true of the flow market, when restricted to one source and many sinks, and the corresponding nonlinear convex program. This fact and the striking resemblance between the Eisenberg–Gale program and Kelly's program naturally raise the question of a systematic study of convex programs having same form as Eisenberg–Gale programs and their corresponding markets. The latter include all markets in Kelly's resource allocation framework.

In this paper we formally define such a class of markets, the *Eisenberg–Gale markets* (*EG markets*). In order to simplify the picture, we define an abstract market corresponding to each EG market; the transformation of prices between the original market and the abstract market is straightforward. By imposing conditions on the inequalities defining the abstract market, we identify natural subclasses of EG markets—SUA and UUA markets (see Section 6). These in turn help us obtain general algorithmic and game-theoretic results; the three game-theoretic properties we consider are efficiency, fairness, and competition monotonicity. We also study rationality of equilibria. Finally, we derive insights into the algorithmic and game-theoretic properties of specific resource allocation markets, using the general results wherever possible.

1.1. Game-theoretic properties studied

Our definition of *efficiency* of a market is very much related to, and indeed inspired by, the notion of price of anarchy given by Koutsoupias and Papadimitriou (1999). In a sense, we wish to study the "price of capitalism" of a market: In the presence of multiple agents who have different abilities to control the market and have their own idiosyncratic ways of utilizing the limited resources available, how efficient is the market, in terms of overall output, once a pricing mechanism has forced it into an equilibrium? We define the efficiency of a market as the worst possible ratio of equilibrium utilities and the centralized optimum solution.

We define *fairness* in terms of the notion of bang-per-buck, i.e., utility received per unit money spent, of agents. For SUA markets we show that equilibria satisfy both max-min and min-max fairness w.r.t. bang-per-buck. Max-min and min-max fairness are considered the strongest notions of fairness in game theory.

Markets satisfying the property of *weak gross substitutability* (raising the price of one good does not decrease the demand for another good) have been extensively studied before. In our abstract markets, we have dispensed with goods and we are dealing directly with utility. Is there a replacement for the fundamental property of weak gross substitutability in this setting?

For this purpose, we define the notion of *competition monotonicity*. We say that a market satisfies this condition if increasing the money of one agent cannot lead to an increase in the utility of any other agent. Observe that after the money

[2] In particular, Kelly's object was to explain the unprecedented success of TCP, and its congestion avoidance protocol due to Jacobson (1988), which played a crucial role in the phenomenal growth of the Internet and the deployment of a myriad of diverse applications on it. Fairness is a key property desired of a congestion avoidance protocol and Jacobson's protocol does seem to ensure fairness. Recent results show that if Jacobson's protocol is run on the end-nodes and the Floyd–Jacobson protocol (Floyd and Jacobson, 1993) is run at buffer queues, in the limit, traffic flows converge to an optimal solution of Kelly's convex program, i.e., they are equilibrium allocations, see Low and Lapsley (1999), Low et al. (2002), Low (2003). Furthermore, Kelly used his convex programming formulation to prove that equilibrium allocations in his model satisfy proportional fairness (see Section 3), thereby giving a formal ratification of Jacobson's protocol.

of one agent is increased, the rest of the agents need to compete against a more powerful competitor. Hence competition monotonicity is a reasonable property to expect in a market. We prove that any Fisher market satisfying weak gross substitutability must also satisfy competition monotonicity. Using our general algorithmic result for SUA markets, together with cross-monotonic cost allocations for submodular cost functions given in Jain and Vazirani (2002), we show that all SUA markets satisfy competition monotonicity.

1.2. Algorithmic contributions

Similar to Devanur et al. (2008), our algorithms also use the primal-dual schema in the enhanced setting of convex programming and KKT conditions, rather than its usual setting of linear programming duality. Our algorithm for the single-source multiple-sink market has a simple description as an ascending price auction in which buyers are sinks and sellers are edges. Buyers keep increasing their bids; since they have fixed budgets, their demands keep decreasing. As soon as a minimum cut realizes that any more decrease in demand will leave it under-utilized, it sells itself and fixes its price. All buyers, whose demand cannot be decreased without making this cut under-saturated, freeze their bids and demands; the remaining buyers keep increasing their bids. The only conceptual difference from a real auction situation is that in our algorithm, sellers are highly coordinated.

Several primal-dual algorithms for solving linear programs are best viewed as ascending price auctions, e.g. Jain and Vazirani (2001). Our algorithm, operating within the framework of a convex program and KKT conditions, is fundamentally different in the following respects. Complementary slackness conditions of LP's are equations that involve only primal variables or only dual variables. On the other hand, equations in the KKT conditions involve both primal and dual variables. This difference manifests itself as follows. In LP-based algorithms, buyers have fixed demands and are trying to minimize the price at which they buy; in our problem, buyers have fixed budgets and they are trying to minimize the rate at which they buy and hence maximize the amount they buy.

In Section 7 we give a strongly polynomial, primal-dual algorithm for SUA markets which can also be viewed as an ascending price auction. We note that Groenevelt (1991) has given a polynomial time algorithm for maximizing a separable concave function over a polymatroid feasible region which, when run with the recent strongly polynomial algorithms of Fleischer et al. (2001), Schrijver (2000) for minimizing a submodular function, gives a strongly polynomial algorithm for solving the Eisenberg–Gale-type convex program corresponding to an SUA market. However, since our algorithm is explicitly seeking an equilibrium, rather than an optimal solution to this convex program, it may be more relevant to an algorithmic theory of equilibria.

In a recent result, Nagano (2007) shows that finding the lexicographically optimal base in a polymatroid region is equivalent to finding the equilibrium of an SUA market.

2. Fisher's model and the Eisenberg–Gale program

In this paper, we will define Fisher markets in a more general way than intended by Fisher (Brainard and Scarf, 2000). Consider a market consisting of a set of n buyers (or agents), $A = \{1, 2, \dots, n\}$, and a set of k indivisible goods, B. Let $m_1, \dots, m_n$ be the money possessed by the n buyers. Corresponding to each buyer $i \in A$ is a monotonically increasing function $u_i : \mathbf{R}_+^k \to \mathbf{R}_+$ which gives the utility derived by agent i on receiving an allocation of goods. The problem is to find prices $p_1, \dots, p_k$ for the goods so that when each buyer is given her utility maximizing bundle of goods, the market clears, i.e., each good having a positive price is exactly sold, i.e., without there being any deficiency or surplus. Since the u_i's are monotonically increasing, all the money of the agents is also fully used up. Such prices are called *market clearing prices*.

The linear utilities case of Fisher's market model is especially important and well studied. For this case, we may assume w.l.o.g. that we have a unit amount of each good. Let u_{ij} be the utility derived by buyer i on receiving one unit of good j. In order to ensure that all goods are sold off at equilibrium, we make the (mild) assumption that for each good j there is a buyer i such that $u_{ij} > 0$. This condition ensures that the equilibrium price of each good is positive (if not, some buyer will demand an infinite amount of that good). Let x_{ij} denote the amount of good j that buyer i gets. Then the total utility derived by i is

$$u_i = \sum_{j=1}^{k} u_{ij} x_{ij}.$$

The problem again is to find market clearing prices.

Clearly, a convex program whose optimal solution is an equilibrium allocation must have as constraints the packing constraints on the x_{ij}'s. Furthermore, its objective function, which attempts to maximize utilities derived, should satisfy:

- If the utilities of any buyer are scaled by a constant, the optimal allocation remains unchanged.
- If the money of a buyer b is split among two new buyers whose utility functions are the same as that of b then sum of the optimal allocations of the new buyers should be an optimal allocation for b.

The money weighted geometric mean of buyers' utilities satisfies both these conditions:

$$\max\left(\prod_{i\in A} u_i^{m_i}\right)^{1/\sum_i m_i}.$$

Clearly, the following objective function is equivalent:

$$\max \prod_{i\in A} u_i^{m_i}.$$

Its log is used in the Eisenberg–Gale convex program:

$$\begin{aligned}
&\text{maximize} \quad \sum_{i\in A} m_i \log u_i \\
&\text{subject to} \quad \forall i \in A\colon \quad u_i = \sum_{j=1}^{n} u_{ij} x_{ij}, \\
&\qquad \forall j \in B\colon \quad \sum_{i\in A} x_{ij} \leqslant 1, \\
&\qquad \forall i \in A,\ j \in B\colon \quad x_{ij} \geqslant 0.
\end{aligned} \tag{1}$$

Interpret Lagrangian variables, say p_i's, corresponding to the second set of conditions as prices of goods. Using KKT conditions one can show that optimal solutions to x_{ij}'s and p_j's give an equilibrium for Fisher's problem (for a proof, see e.g., Theorem 1 in Vazirani, 2007). Using the fact that the objective function of this convex program is strictly concave, it is easy to show that the utility derived by each buyer is the same in all equilibrium allocations and the equilibrium prices are unique.

3. Resource allocation markets

Kelly considered the following general setup for modeling resource allocation. Let R be a finite set of resources and $c: R \to \mathbf{Z}^+$ be the function specifying the available capacity of each resource $r \in R$. Let $A = \{a_1, \ldots, a_n\}$ be a set of agents and $m_i \in \mathbf{Z}^+$ be the money available with agent a_i.

Each agent wants to build as many *objects* as possible using resources in R. An agent may be able to use several different subsets of R to make one object. Let $S_{i1}, S_{i2}, \ldots, S_{ik_i}$ be subsets of R usable by agent a_i, $k_i \in \mathbf{Z}^+$. For each agent $a_i \in A$ and index j s.t. $1 \leqslant j \leqslant k_i$, denote by x_{ij} the number of objects a_i makes using the subset S_{ij}; x_{ij} is not required to be integral. Let $f_i = \sum_{j=1}^{k_i} x_{ij}$ be the total number of objects made by agent a_i. We will say that x is *feasible* if in building the corresponding objects, the capacity constraints, c, on R are not violated. Similarly, we will say that f_i, $1 \leqslant i \leqslant n$, is *feasible* if simultaneously each agent a_i can make f_i objects without violating capacity constraints on R.

We will view the setup given above as a market and will consider the problem of finding prices for resources in R and a feasible x such that they are in *equilibrium* in the following sense:

1. Resource $r \in R$ has positive price only if it is used to capacity under x.
2. Each agent uses only the cheapest sets to make objects.
3. The money of each agent is fully spent.

Interestingly enough, in this resource allocation context, Kelly considered the following convex program which has the same form as the Eisenberg–Gale program. Using KKT conditions, one can show that an optimal solution to this convex program is an equilibrium solution (for a proof, see e.g., Theorem 1 in Vazirani, 2007).

$$\begin{aligned}
&\text{maximize} \quad \sum_{a_i\in A} m_i \log f_i \\
&\text{subject to} \quad \forall a_i \in A\colon \quad f_i = \sum_{j=1}^{k_i} x_{ij}, \\
&\qquad \forall r \in R\colon \quad \sum_{(ij)\colon\ r\in S_{ij}} x_{ij} \leqslant c(r), \\
&\qquad \forall a_i \in A,\ 1 \leqslant j \leqslant k_i\colon \quad x_{ij} \geqslant 0.
\end{aligned} \tag{2}$$

Let p_r, $r \in R$, be Lagrangian variables corresponding to the second set of conditions; we will interpret these as prices of resources. By the KKT conditions optimal solutions to x_{ij}'s and p_r's must satisfy:

1. $\forall r \in R$: $p_r \geqslant 0$.
2. $\forall r \in R$: $p_r > 0 \Rightarrow \sum_{(ij):\ r \in S_{ij}} x_{ij} = c(r)$.
3. $\forall a_i \in A$, $1 \leqslant j \leqslant k_i$: $\frac{m_i}{f_i} \leqslant \sum_{r \in S_{ij}} p_r$.
4. $\forall a_i \in A$, $1 \leqslant j \leqslant k_i$: $x_{ij} > 0 \Rightarrow \frac{m_i}{f_i} = \sum_{r \in S_{ij}} p_r$.

Kelly showed that an optimal solution to program (2), and hence an equilibrium solution to the corresponding market, satisfies *proportional fairness*, i.e., if f_i^* is an optimal solution and f_i is any feasible solution, then

$$\sum_{i=1}^{n} \frac{f_i - f_i^*}{f_i^*} \leqslant 0.$$

Intuitively, the only way of making an agent happier by 5% is to make other agents unhappy by at least a total of 5%.

This general setup can be used to model many situations; in this paper we will consider the following combinatorial markets. The first two of these were especially relevant to Kelly's work and the third and fourth model situations in which sources need to buy capacity to broadcast to all nodes in the network.

1. **Market 1 (Flow market):** Given a finite, directed graph $G = (V, E)$, E is the set of resources, with capacities specified. Agents are source-sink pairs of nodes, $(s_1, t_1), \ldots, (s_k, t_k)$, with money $m_1, \ldots, m_k$, respectively. Each $s_i - t_i$ path is an object for agent (s_i, t_i).
2. **Market 2 (Flow market):** Same as above, except the graph is undirected.
3. **Market 3:** Given a directed graph $G = (V, E)$, E is the set of resources, with capacities specified. Agents are $A \subset V$, each with specified money. For $s \in A$ objects are branchings rooted at s and spanning all V, i.e., directed trees rooted at s and containing a path from s to each vertex in V.
4. **Market 4:** Same as above, except the graph is undirected and objects are spanning trees.

4. Algorithm for single-source multiple-sink markets

In this section, we consider the special case of Markets 1 and 2 with a single source and multiple sinks. We will assume that the underlying graph is directed. In case it is undirected, one can use the standard reduction from undirected graphs to directed graphs—replace each undirected edge (u, v) with the two edges (u, v) and (v, u) of the same capacity.

Formally, let $G = (V, E)$ be a directed graph with capacities on edges. Let $s \in V$ be the source node and $T = \{t_1, \ldots, t_r\}$ be the set of sink nodes, also called terminals. Let m_i be the money possessed by sink t_i. The problem is to determine equilibrium flow and edge prices. The following example may help appreciate better some of the intricacies of this problem.

Example 1. Consider graph $G = (V, E)$ with $V = \{s, a, b, c, d\}$ and sinks b and d with \$120 and \$10, respectively. The edges are: (s, a), (s, c) having capacity 2, (a, b) having capacity 1, and (a, d), (c, d), (c, b) having capacity 10. The unique equilibrium prices are $p_{(s,a)} = \$10$, $p_{(a,b)} = \$30$, $p_{(s,c)} = \$40$, and the rest of the edges have zero price. At equilibrium, flow on path s, a, d is one, on s, a, b is one, and on s, c, b is two. Simulating the algorithm below on this example will reveal the complex sequence of cuts it needs to find in order to compute the equilibrium. Computing equilibrium for other values of money is left as an interesting exercise.

We will present a strongly polynomial algorithm for this problem which is based on the primal-dual schema, i.e., it alternately adjusts flows and prices, attempting to satisfy all KKT conditions. Often, primal-dual algorithms can naturally be viewed as executing an auction. This viewpoint leads to a particularly simple way of presenting the current algorithm. We will describe it as an ascending price auction in which the buyers are sinks and sellers are edges. The buyers have fixed budgets and are trying to maximize the flow they receive and the sellers are trying to extract as high a price as possible from the buyers.

One important deviation from the usual auction situation is that the sellers act in a highly coordinated manner—at any point in the algorithm, all edges in a particular cut, say $(S, \overline{S})$, raise their prices simultaneously while prices of the remaining edges remain unchanged (note that the *cut* consists of all edges (u, v) with $u \in S$ and $v \in \overline{S}$). The prices of all edges are initialized to zero. The first cut considered by the algorithm is the (unique) maximal min-cut separating all sinks from s, say $(S_0, \overline{S_0})$.

Denote by $\text{rate}(t_i)$ the cost of the cheapest $s - t_i$ path w.r.t. current prices. The flow demanded by sink t_i at this point is $m_i/\text{rate}(t_i)$. At the start of the algorithm, when all edge prices are zero, each sink is demanding infinite flow. Therefore, the algorithm will not be able to find a feasible flow that satisfies all demands. Indeed, this will be the case all the way until termination; at any intermediate point, some cuts will need to be over-saturated in order to meet all the demand.

The price of edges in cut $(S, \overline{S})$ is raised as long as the demand across it exceeds supply, i.e., the cut is over-saturated due to flow demanded by sinks in $\overline{S}$. At the moment that demand exactly equals supply, the edges in this cut stop raising prices and declare themselves sold at current prices. This makes sense from the viewpoint of the edges in the cut—if they raise prices any more, demand will be less than supply, i.e., the cut will be under-saturated, and then these edges will have to be priced at zero!

The crucial question is when does the cut $(S, \overline{S})$ realize that it needs to sell itself? This point is reached as soon as there is a cut, say $(U, \overline{U})$, with $S \subset U$, such that the difference in the capacities of the two cuts is precisely equal to the flow demanded by sinks in $\overline{S} - \overline{U}$. Let $(U, \overline{U})$ be the maximal such cut (it is easy to see that it will be unique). If $U = V$, the algorithm halts. Otherwise, cut $(U, \overline{U})$ must be over-saturated—it assumes the role of $(S, \overline{S})$ and the algorithm goes to the next iteration.

Note that an edge may be present in more than one cut whose price is raised by the algorithm. If so, its price will be simply the sum of the prices assigned to these cuts.

Suppose the algorithm executes k iterations. Let $(S_i, \overline{S_i})$ be the cut it finds in iteration i, $1 \leqslant i \leqslant k$, with $S_k = V$. Clearly, we have $S_0 \subset S_1 \subset \cdots \subset S_k = V$. Let T_i be the set of terminals in $S_i - S_{i-1}$, for $1 \leqslant i \leqslant k$. Let c_i be the set of edges of G in the cut $(S_i, \overline{S_i})$, for $0 \leqslant i < k$ and p_i be the price assigned to edges in c_i. Clearly, for each terminal $t \in T_i$, $\text{rate}(t) = p_0 + \cdots + p_{i-1}$, for $1 \leqslant i \leqslant k$.

Let G' denote the graph obtained by adding a new sink node t to G and edges (t_i, t) from each of the original sinks to t. Let the capacity of edge (t_i, t) be $m_i/\text{rate}(t_i)$. For convenience, even in G', we will denote $V - S$ by $\overline{S}$. It is easy to see that each of the cuts $(S_i, \overline{S_i} \cup \{t\})$ in G' has the same capacity, for $0 \leqslant i \leqslant k$, and each of these $k+1$ cuts is a minimum $s - t$ cut in G'.

Let f' denote a maximum $s - t$ flow in G'. Obtain flow f from f' by ignoring flow on the edges into t. Then f is a feasible flow in G that sends $m_i/\text{rate}(t_i)$ flow to each sink t_i.

Lemma 2. *Flow f and the prices found by the algorithm constitute an equilibrium flow and prices.*

Proof. We will show that flow f and the prices found satisfy all KKT conditions.

- Since each of the cuts $(S_i, \overline{S_i} \cup \{t\})$, for $0 \leqslant i < k$ is saturated in G' by flow f', each of the cuts $c_0, c_1, \ldots, c_{k-1}$ is saturated by f. Hence, all edges having non-zero prices must be saturated.
- The cost of the cheapest path to terminal $t' \in T$ is $\text{rate}(t')$. Clearly, every flow to t' uses a path of this cost.
- Since the flow sent to $t' \in T$ is $m_i/\text{rate}(t')$, the money of each terminal is fully spent. $\square$

Below we give a strongly polynomial time subroutine for computing the next cut in each iteration.

4.1. Finding the next cut

In this subsection we will use ideas from Megiddo's work (Megiddo, 1974), giving an algorithm for finding a max-min fair flow in a single-source multiple-sink network. Let $(S, \overline{S})$ be the cut in G whose price is being raised in the current iteration and let c be the set of edges in this cut and f its capacity. Let T' denote the set of sinks in $\overline{S}$. Let p' denote the sum of the prices assigned to all cuts found so far in the algorithm (this is a constant for the purposes of this subroutine) and let p denote the price assigned to edges in c. The cut $(S, \overline{S})$ satisfies the following conditions:

- It is a maximal minimum cut separating T' from s.
- At $p = 0$, every cut $(U, \overline{U})$, with $S \subseteq U$, is over-saturated.

Let p^* be the smallest value of p at which there is a cut $(U, \overline{U})$, with $S \subset U$, in G such that the difference in the capacities of $(S, \overline{S})$ and $(U, \overline{U})$ is precisely equal to the flow demanded by sinks in $U - S$ at prices p^*; moreover, $(U, \overline{U})$ is the maximal such cut. Below we give a strongly polynomial algorithm for finding p^* and $(U, \overline{U})$.

Define graph G' by adding a new sink node t to G and edges (t_i, t) for each sink $t_i \in \overline{S}$. Define the capacity of edge (t_i, t) to be $m_i/(p' + p)$ where m_i is the money of sink t_i. As in Section 4 we will denote $V - S$ by $\overline{S}$ even in G'. The proof of the following lemma is obvious.

Lemma 3. *At the start of the current iteration, $(S, \overline{S} \cup \{t\})$ is a maximal minimum $s - t$ cut in G'. p^* is the smallest value of p at which a new minimum $s - t$ cut appears in G'. $(U, \overline{U} \cup \{t\})$ is the maximal minimum $s - t$ cut in G' at price p^*.*

For any cut C in G', let $\text{cap}_p(C)$ denote its capacity, assuming that the price of each edge in c is p. For $p \geqslant 0$, define $\text{cut}(p)$ to be the maximal $s - t$ min-cut in G' assuming that the price assigned to edges in c is p. For cut $(A, \overline{A} \cup \{t\})$, $A \subseteq V$, let $\text{price}(A, \overline{A} \cup \{t\})$ denote the smallest price that needs to be assigned to edges in c to ensure that $\text{cap}_p(A, \overline{A} \cup \{t\}) = f$, i.e., $(A, \overline{A} \cup \{t\})$ is also a min $s - t$ cut in G'; if $(A, \overline{A} \cup \{t\})$ cannot be made a minimum $s - t$ cut for any price p then $\text{price}(A, \overline{A} \cup \{t\}) = \infty$. Clearly, $\text{price}(A, \overline{A} \cup \{t\}) \geqslant p^*$. Observe that determining $\text{price}(A, \overline{A} \cup \{t\})$ involves simply solving an equation in which p is unknown.

Lemma 4. *Suppose $p > p^*$. Let* $\text{cut}(p) = (A, \overline{A} \cup \{t\})$, *where $A \neq U$. Let* $\text{price}(A, \overline{A} \cup \{t\}) = q$ *and* $\text{cut}(q) = (B, \overline{B} \cup \{t\})$. *Then $B \subset A$.*

Subroutine
Inputs: Cut $(S, \overline{S})$ in G whose price is being raised in the current iteration.
Output: Price p^* and next cut $(U, \overline{U})$.

1. $C \leftarrow (V, t)$
2. $p \leftarrow \text{price}(C)$
3. While $\text{cut}(p) \neq C$ do:
   (a) $C \leftarrow \text{cut}(p)$
   (b) $p \leftarrow \text{price}(C)$
4. Output (C, p)

Fig. 1. Subroutine for finding next cut.

Proof. Since we have assumed that $A \neq U$, it must be the case that $\text{cap}_p(A, \overline{A} \cup \{t\}) > f$. Therefore, $q = \text{price}(A, \overline{A} \cup \{t\}) < p$. Let c_A and c_B denote the capacities of $(A, \overline{A} \cup \{t\})$ and $(B, \overline{B} \cup \{t\})$ at price $p = 0$. Let m_A and m_B denote the money possessed by sinks in $(A - S)$ and $(B - S)$, respectively.

Since $(A, \overline{A} \cup \{t\})$ is a maximal $s - t$ min-cut at price p,

$$c_A + \frac{m_A}{p} < c_B + \frac{m_B}{p}.$$

Since $(B, \overline{B} \cup \{t\})$ is a maximal $s - t$ min-cut at price q,

$$c_B + \frac{m_B}{q} < c_A + \frac{m_A}{q}.$$

The two together imply

$$\frac{m_B - m_A}{q} < c_A - c_B < \frac{m_B - m_A}{p}.$$

First suppose that $A \subset B$. Clearly $m_A \leqslant m_B$. But this contradicts the last inequality since $q < p$.

Next, suppose that A and B cross. By the last inequality above, there must be a price, r, such that $q < r < p$ at which $\text{cap}_r(A, \overline{A} \cup \{t\}) = \text{cap}_r(B, \overline{B} \cup \{t\}) = g$, say. By the submodularity of cuts, one of the following must hold:

1. $\text{cap}_r((A \cap B), \overline{(A \cap B)} \cup \{t\}) \leqslant g$. Since the money possessed by sinks in $(A \cap B) - S$ is at most m_B, at price q, $\text{cap}_q((A \cap B), \overline{(A \cap B)}\{t\}) < \text{cap}_q(B, \overline{B} \cup \{t\})$. This contradicts the fact that $(B, \overline{B} \cup \{t\})$ is a min-cut at price q.
2. $\text{cap}_r((A \cup B), \overline{(A \cup B)} \cup \{t\}) \leqslant g$. Since the money possessed by sinks in $(A \cup B) - S$ is at least m_A, at price p, $\text{cap}_p((A \cup B), \overline{(A \cup B)} \cup \{t\}) < \text{cap}_p(A, \overline{A} \cup \{t\})$. This contradicts the fact that $(A, \overline{A} \cup \{t\})$ is a min-cut at price p.

Hence we get that $B \subset A$. $\square$

Lemma 5. *Subroutine in Fig.* 1 *terminates with the cut $(U, \overline{U} \cup \{t\})$ and price p^* in at most r max-flow computations, where r is the number of sinks.*

Proof. As long as $p > p^*$, by Lemma 4, the algorithm keeps finding smaller and smaller cuts, containing fewer sinks on the s side. Therefore, in at most r iterations, it must arrive at a cut such that $p = p^*$. Since $\text{cut}(p^*) = (U, \overline{U} \cup \{t\})$, the next cut it considers is $(U, \overline{U} \cup \{t\})$. Since $\text{price}(U, \overline{U} \cup \{t\}) = p^*$, at this point the algorithm terminates. $\square$

Theorem 6. *The algorithm given in Section* 4 *finds a equilibrium edge prices and flows using $O(r^2)$ max-flow computations, where r is the number of sinks.*

Proof. Clearly, the number of sinks trapped in the sets $S_0 \subset S_1 \subset \cdots \subset S_k$ keeps increasing and therefore, the number of iterations $k \leqslant r$. The running time for each iteration is dominated by the time taken by subroutine from Fig. 1, which by Lemma 5 is r max-flow computations. Hence the total time taken by the algorithm is $O(r^2)$ max-flow computations. By Lemma 2 the flow and prices found by the algorithm are equilibrium flow and prices. $\square$

5. Eisenberg–Gale markets

The striking resemblance between the Eisenberg–Gale program and Kelly's program (2), and the fact that solutions to the former are rational, and so are solutions to the latter for the case of single-source multiple-sink markets, naturally raises the question of a systematic study of convex programs having the same form as Eisenberg–Gale programs and their corresponding markets.

Over the years, convex programs with the same basic structure as the Eisenberg–Gale program were studied for Fisher's model under the following utility functions: scalable utilities (Eisenberg, 1961), Leontief utilities (Codenotti and Varadarajan, 2004), and homothetic utilities with productions (Jain et al., 2005). In this section, we will define a market model that generalizes all these. In particular, this model and its subclasses will make it easier to understand several natural resource allocation markets defined in Section 3.

We first define the notion of an *Eisenberg–Gale-type convex program*. This class includes any convex program whose objective function is of the form

$$\sum_{i\in A} m_i \log u_i$$

subject to linear packing constraints (in which all coefficients are non-negative), where m_i's are constants and u_i's are variables which get constrained in the packing constraints. In addition, any convex program that can be transformed, for instance, by eliminating some variables or via the Fourier–Motzkin elimination method, into a program of this form is also an Eisenberg–Gale-type convex program.

Let $\mathcal{M}$ be a Fisher market whose set of feasible allocations and buyers' utilities is captured by a polytope P. The linear constraints defining P contain two types of variables, utility variables and allocation variables. We will use index i to vary over variables and index j to vary over constraints and will assume w.l.o.g. that the first n variables represent buyers' utilities and the rest represent allocations. We will further assume that the linear constraints describing P are packing constraints, i.e., they can be written as $\sum_i a_{ij} x_i \leqslant b_j$, with all coefficients being non-negative. Observe that such a market satisfies the *free disposal property*, i.e., whenever $\boldsymbol{u}$ is a feasible utility vector then so is any vector dominated by $\boldsymbol{u}$.

Let $f_i : \mathbf{R}_+ \to \mathbf{R}_+$ be a strictly monotonic transformation, for $1 \leqslant i \leqslant n$. Then, replacing the utility functions u_i by $f_i(u_i)$ will yield a market that is equivalent to $\mathcal{M}$. This is so because in a Fisher market, the exact utility function specified is not really important; what is important is the underlying relation defined over bundles of goods via this utility function. Given two bundles of goods, this relation tells us whether agent i is equally happy with both these bundles or whether one bundle makes i happier than the other bundle. Clearly, this relation remains unchanged if utility functions u_i are replaced by the functions $f_i(u_i)$.

We will say that an allocation $x_1, \dots, x_n$ made to the buyers is a *clearing allocation* if it uses up all goods exactly to the extent to which they were available in market $\mathcal{M}$.

We will say that Fisher market $\mathcal{M}$ satisfying the conditions stated above is an *Eisenberg–Gale market* if there exist strictly monotonic transformations $f_i : \mathbf{R}_+ \to \mathbf{R}_+$, for $1 \leqslant i \leqslant n$, such that any clearing allocation $x_1, \dots, x_n$ which maximizes

$$\sum_{i\in A} m_i \log f_i\bigl(u_i(x_i)\bigr)$$

is an equilibrium allocation, i.e., there exist prices $p_1, \dots, p_k$ for the goods such that for each agent i, x_i is a utility maximizing bundle relative to these prices.

It is easy to see, using the KKT conditions, that equilibrium allocations for an Eisenberg–Gale market are captured as an optimal solution to an Eisenberg–Gale-type convex program.

All markets studied in this paper are simple enough that picking f_i's to be the identity transformation suffices. The market considered in Jain et al. (2005), under homothetic quasi-concave utility functions, requires non-trivial transformations for deducing that in fact it is an Eisenberg–Gale market. Since the equilibrium of an EG market is captured via a convex program, equilibrium utilities can be approximated to any required degree in polynomial time (though not strongly polynomial time) using the ellipsoid algorithm.

Remark 7. Observe that even if the number of constraints defining $\mathcal{M}$ are not finite in number, its equilibrium could still be captured via an Eisenberg–Gale-type convex program. We have restricted ourselves to finitely many constraints for the sake of simplicity.

In order to simplify the picture, let us project polytope P onto the utility variables to get polytope P_u. Assume that the constraints of P_u are written in the following form

$$\forall q: \quad \sum_p c_{pq} u_p \leqslant d_q,$$

where u_p is buyer p's utility and index q varies over constraints. Because of the free disposal assumption, $c_{pq} \geqslant 0$, and because utilities are non-negative, $d_q \geqslant 0$. These constraints define the Eisenberg–Gale-type convex program of this market.

In this abstract market, there are no goods. However, we may view each constraint q as defining an abstract good which is desired by only the buyers having non-zero coefficients in this constraint. A set of linear constraints representing P_u can be obtained using the Fourier–Motzkin elimination method (see Schrijver, 1986). Therefore, the constraints describing P_u are linear combinations of the constraints used for describing P. Hence the dual variables of the abstract market (which correspond to constraints of P_u) are the same linear combination of the dual variables of the concrete market (which correspond to the constraints of P). This shows how to transform prices from the abstract market to the concrete market. Clearly, the reverse direction is also straightforward.

The simplicity of the abstract markets helps us obtain a general algorithmic result as well as uncover several game-theoretic properties. In the next section, we use this abstraction to define subclasses of the class of Eisenberg–Gale markets and in Section 7 we present a strongly polynomial algorithm for finding equilibria for all markets in one of these subclasses, SUA markets. This algorithm will in turn easily yield algorithms for some of the resource allocation markets defined in Section 3. Other properties of these classes will be studied in Sections 11–13 and 14.

6. UUA and SUA markets

Assume that an abstract market is defined by packing constraints of the form:

$$\forall j:\quad \sum_i a_{ij}u_i \leqslant b_j,$$
$$\forall i \in A:\quad u_i \geqslant 0,$$

where a_{ij}'s and b_j's are constants satisfying $a_{ij} \geqslant 0$ and $b_j > 0$. In these constraints, b_j's encode the supply of resources, and for a fixed j, the different a_{ij}'s encode the different ways in which agents use resources. Let us consider the special case in which for each j, all agents interested in this resource use it in the same way, i.e., all a_{ij}'s are restricted to be 0/1. Clearly, there can be at most 2^n such relevant constraints. The b_j's for these constraints can be encoded via a function $v : 2^A \to \mathbf{R}_+$ and the constraints can then be written as:

$$\forall S \subseteq A:\quad \sum_{i\in S} u_i \leqslant v(S),$$
$$\forall i \in A:\quad u_i \geqslant 0.$$

We will say that a utility vector $\boldsymbol{u}$ is *feasible for* v if it satisfies all these constraints. For set $S \subseteq A$ we will say that $\boldsymbol{u}$ makes S *tight* if $\sum_{i\in S} u_i = v(S)$. Let us define the *covering closure*, v^*, of v as follows. For $S \subseteq A$, define $v^*(S)$ to be the minimum cost fractional covering of S, i.e., the solution to the following linear program:

$$\min \sum_{T\subseteq A} v(T)x_T,$$
$$\forall i \in S:\quad \sum_{T:\ i\in T} x_T \geqslant 1,$$
$$\forall T \subseteq A:\quad x_T \geqslant 0.$$

A function whose covering closure is itself (e.g., v^*), will be said to satisfy the *covering property*. Clearly, such a function must satisfy *monotonicity*, i.e., $\forall S \subseteq T:\ v^*(S) \leqslant v^*(T)$. Clearly, a utility vector $\boldsymbol{u}$ is feasible for v iff it is feasible for v^*. Therefore, we will assume w.l.o.g. that function $v : A \to \mathbf{R}_+$ satisfies the covering property, monotonicity and the fact that $v(\emptyset) = 0$.

Given such a function v, let us define a market $\mathcal{M}(v)$ as follows. An *instance of* $\mathcal{M}(v)$ is specified by specifying the money possessed by each agent. Let m_i be the money possessed by each agent i under an instance I. The object is to find utilities of agents, $\boldsymbol{u}$, and prices p_S for each subset $S \subseteq A$ such that:

1. $\forall S \subseteq A:\ p_S \geqslant 0$,
2. $\forall S \subseteq A:\ p_S > 0 \Rightarrow$ $\boldsymbol{u}$ makes S tight,
3. $\forall i \in A:\ \frac{m_i}{u_i} = \text{rate}(i)$,

where for $i \in A$, rate(i) is defined to be the sum of prices of all sets containing i, i.e.,

$$\text{rate}(i) = \sum_{S:\ i\in S} p_S.$$

This is the rate at which agent i gets one unit of utility. The traditional notion of *bang-per-buck* of agent i is the reciprocal of rate(i); it is the number of units of utility that i derives per unit of money. Thus, the bang-per-buck of agent i is u_i/m_i.

We will call such a market a *uniform utility allocation market* and will abbreviate it to *UUA market*. An example of such a market is Fisher's model under the restriction that all u_{ij}'s are 0/1.

Consider the following convex program corresponding to instance I of market $\mathcal{M}(v)$.

$$\text{maximize}\quad \sum_{i\in A} m_i \log u_i \tag{3}$$
$$\text{subject to}\quad \forall S \subseteq A:\quad \sum_{i\in S} u_i \leqslant v(S),$$
$$\forall i \in A:\quad u_i \geqslant 0.$$

Let p_S, $S \subseteq A$, be Lagrangian variables corresponding to the first set of conditions. As in the case of the Eisenberg–Gale program, using KKT conditions, one can show that optimal solutions to $\boldsymbol{u}$ and $\boldsymbol{p}$ must give equilibrium utilities and prices for the market.

Function v is said to be a *submodular function* if

$$v(S) + v(T) \geqslant v(S \cap T) + v(S \cup T),$$

for any sets $S, T \subseteq A$. It is easy to see that a submodular function satisfies the covering property. Function v is said to be a *monotone* if $v(S) \leqslant v(T)$ for $S \subseteq T \subseteq A$.

An important special case arises when v is a *polymatroid function*, i.e., it is submodular, monotone, and $v(\emptyset) = 0$. In this case we will say that $\mathcal{M}(v)$ is a *submodular utility allocation market* and will abbreviate it to *SUA market*. Fisher's model under the restriction that all u_{ij}'s are 0/1 is an example of such a market. Megiddo (1974) showed that in a single-source multiple-sink network, the function specifying the total flow possible to a subset of sinks is submodular. As a result, the single-source multiple-sink market is also an SUA market.

7. Algorithm for SUA markets

Let $\mathcal{M}(v)$ be an SUA market. We will first use the submodularity of v to define a canonical dual solution. We may assume w.l.o.g. that under instance I, $M(I) = A$, i.e., all agents have positive money.

Two sets $S, T \subset A$ are said to be *uncrossed* if they are either disjoint or one is contained in the other. If S, T are not uncrossed, they are said to be *crossing sets*. Let $\boldsymbol{u}$ be feasible for v. Using the submodularity of v it is easy to see that if S and T are tight then so are $S \cap T$ and $S \cup T$. Now, by standard uncrossing arguments one can show.

Lemma 8. *For any instance of an SUA market $\mathcal{M}(v)$, equilibrium prices can be chosen in such a way that sets having positive prices form a nested family.*

As stated in the Introduction, our algorithm is best viewed as an ascending price auction. Agent keep raising their bids (rates). When a set realizes that any more increase will leave it under-saturated, it declares itself sold and fixes its price, and agents whose bids cannot be increased any more freeze their bids and demands.

Our algorithm initializes prices of all sets to zero and the rates of all agents to zero. It iteratively finds a nested family of sets, $A = A_1 \supset A_2 \supset \cdots \supset A_k \supset A_{k+1} = \emptyset$ and computes their prices. In the first iteration, it computes the price of A and finds the next set A_2. In general, in iteration j, it computes the price of A_j and finds the next set, A_{j+1}. Clearly, the rates of all agents in $A_j - A_{j+1}$ are the same.

Consider iteration j, $1 \leqslant j \leqslant k$. Let $p_1, \ldots, p_{j-1}$ be the prices assigned to sets $A_1, \ldots, A_{j-1}$ so far. Let $r = p_1 + \cdots + p_{j-1} + p$, where p is a variable whose value will be found in the current iteration. It will eventually be the price of A_j. Price p is initialized to zero and is raised uniformly as described below; as p increases, r also increases.

Define function $h : 2^{A_j} \to \mathbf{R}_+$ as follows:

$$h(S) = v(S) + \frac{m(A_j - S)}{r},$$

where $S \subseteq A_j$. For $S \subset A_j$, we will say that S is *active* if $h(S) = v(A_j)$. Because of the fact that in the previous iteration a minimal active set was found, at the beginning of the current iteration, when $p = 0$, for each $S \subseteq A_j$, $h(S) > v(A_j)$.

In iteration j, p is raised uniformly until some set becomes active. Observe that this condition must eventually be satisfied, since $S = \emptyset$ is also being considered. We show below how to find the correct value of p, say p^*, at which a set becomes active. Using the submodularity of v we show in Lemma 10 that the minimal active set is unique; let $S \subset A_j$ be this set. The price of A_j is set to p^*, i.e., $p_j = p^*$. Also, for all agents $i \in A_j - S$, rate(i) is set to $p_1 + \cdots + p_{j-1} + p_j$. If $S = \emptyset$, the algorithm halts. Otherwise, it sets A_{j+1} to S and it goes to the next iteration.

We summarize the algorithm below.

Algorithm 9 *(Algorithm for an SUA market).*

1. $T \leftarrow A$
2. $r \leftarrow 0$
3. While $T \neq \emptyset$ do
   $p \leftarrow 0$
   Raise p uniformly until a set becomes active. Let $S \subset T$ be a minimal active set.
   $p_T \leftarrow p$
   $r \leftarrow r + p$
   $\forall i \in T - S$: rate$(i) \leftarrow r$.
   $T \leftarrow S$
4. $\forall i \in A$: $u_i \leftarrow m_i/\text{rate}(i)$.

Submodularity of v easily implies that the intersection of two active sets must be active. Therefore we get:

Lemma 10. *The minimal active set is unique.*

Lemma 11. *The utilities and prices computed by Algorithm* 9 *satisfy all KKT conditions.*

Proof. Suppose the algorithm executes k iterations and finds the sets

$$A = A_1 \supset A_2 \supset \cdots \supset A_k \supset A_{k+1} = \emptyset.$$

We will first show that all these sets must be tight. When the kth iteration terminates, $m(A_k)/(p_1 + \cdots + p_k) = v(A_k)$, i.e., the total utility of agents in A_k equals $v(A_k)$. Hence A_k is tight. When the $(k-1)$st iteration terminates,

$$v(A_k) + m(A_{k-1} - A_k)/(p_1 + \cdots + p_{k-1}) = v(A_{k-1}),$$

i.e., the total utility of agents in $A_{k-1} - A_k$ equals $v(A_{k-1}) - v(A_k)$. Now, since A_k is tight, so in A_{k-1}. Continuing in this manner, an easy induction shows that $A_{k-2}, \ldots, A_1$ must also be tight.

We can now establish the three KKT conditions listed in Section 6. The first is obvious, the second follows from the fact that the algorithm raises prices of sets $A_1, \ldots, A_k$ only, and the third is also obvious since for agent i, u_i is defined to be $m_i/\text{rate}(i)$. $\square$

7.1. Finding the next active set

Let T be the current set being handled by the algorithm. As above, define function $h : 2^T \to \mathbf{R}_+$ as follows:

$$h(S) = v(S) + \frac{m(T - S)}{p' + p},$$

where $S \subseteq T$, p' is the sum of prices of all sets found so far by the algorithm and p is the variable giving the price of T.

As before, for $S \subset T$, we will say that S is *active* if $h(S) = v(T)$. As stated above, at the beginning of the iteration when $p = 0$, for each $S \subseteq A_i$, $h(S) > v(T)$ and hence no set is active. Let p^* be the smallest value of p at which a set becomes active and let S^* be the minimal active set; by Lemma 10 S^* is unique.

For $S \subset T$ define price(S) to be the value of p obtained by solving the equation $h(S) = v(T)$. Clearly, price$(S) \geqslant p^*$. For $p > 0$, let set(p) be the minimal set minimizing $v(S) + m(T - S)/(p' + p)$; by a proof similar to Lemma 10 set(p) is unique.

The procedure given below finds p^* and S^*.

Algorithm 12 *(Subroutine for finding p^* and S^*).*

1. $S \leftarrow \emptyset$
2. $p \leftarrow \text{price}(\emptyset)$
3. While $\text{set}(p) \neq S$ do
   $S \leftarrow \text{set}(p)$
   $p \leftarrow \text{price}(S)$
4. Output S, p

Lemma 13. *Let $p > p^*$, $\text{set}(p) = S$, $\text{price}(S) = q$ and $\text{set}(q) = R$. Then $q < p$ and $R \supset S$.*

Proof. Since $p > p^*$ and $\text{set}(p) = S$, at price p, $h(S) < v(T)$. Now, since $\text{price}(S) = q$, $q < p$.

Suppose $R \subset S$. Let $m(T - S) = a$ and $m(T - R) = a + b$. Since $h(S) = v(T)$ at price p, and $h(R) = v(T)$ at price q,

$$v(S) + \frac{a}{p} = v(R) + \frac{a+b}{q}.$$

Since $\text{set}(p) = S$,

$$v(S) + \frac{a}{p} < v(R) + \frac{a+b}{p}.$$

The two together give

$$v(S) - v(R) = \frac{a+b}{q} - \frac{a}{p} < \frac{a+b}{p} - \frac{a}{p},$$

which contradicts $q < p$.

Next suppose that R crosses S. Clearly, there exists a price t, $q < t < p$, at which $h(S) = h(R)$. By submodularity, at this price either $h(S \cap R) \leqslant h(S)$ or $h(S \cup R) \leqslant h(R)$. In the first event, at price p, $h(S \cap R) < h(S)$, contradicting the fact that $\mathrm{set}(p) = S$. In the second event, at price q, $h(S \cup R) < h(R)$, contradicting the fact that $\mathrm{set}(q) = R$.

Hence, $R \supset S$. □

Corollary 14. *Algorithm* 12 *terminates with p^* and S^* in at most $|T|$ calls to a submodular function minimization oracle.*

Strongly polynomial algorithms for the minimization of a submodular function were given by Fleischer et al. (2001), Schrijver (2000). Since Algorithm 9 finds a nested family of sets, it executes at most n iterations. Now by Lemma 11 and Corollary 14 we get:

Theorem 15. *Algorithm* 9 *computes a market equilibrium for the given submodular utility allocation market in strongly polynomial time.*

8. The structure of Markets 1 and 2

A market having only one agent is clearly an SUA market. We give Example 21 to show that Market 1 is not a UUA market for $k \geqslant 2$, where k is the number of agents (source-sink pairs). For Market 2, we will use Hu's theorem, Theorem 16, to show that it is an SUA market for $k = 2$, and it is not a UUA market for $k \geqslant 3$ (Example 20).

Let F be the maximum flow that can be routed from s_1 to t_1 and s_2 to t_2 simultaneously, i.e., maximize the sum of flow that can be routed subject to edge capacity constraints. Hu shows that F equals the minimum cut separating both pairs of terminals. There are two possibilities for such a cut: it either separates s_1 and s_2 from t_1 and t_2 or it separates s_1 and t_2 from s_2 and t_1. Obtaining min-cuts for both cases and keeping the smaller will give the correct answer.

Let S denote the set of edges in this min-cut; clearly, its capacity is F. Let F_i be the value of max-flow from s_i to t_i and S_i be the set of edges in a min-cut separating s_i and t_i, for $i = 1, 2$. Hu also showed the following theorem which characterizes the region of feasible flow.

Theorem 16. *(See Hu (1963).) Let f_1 and f_2 be two non-negative real numbers such that $f_1 \leqslant F_1$, $f_2 \leqslant F_2$ and $f_1 + f_2 \leqslant F$. Then there exists a feasible flow which carries f_1 amount of flow from s_1 to t_1 and f_2 amount of flow from s_2 to t_2.*

Lemma 17. *Let $v : A \to \mathbf{R}_+$ be a function satisfying the covering property and that $v(\emptyset) = 0$. Then v is submodular.*

Proof. Let $A = \{a, b\}$. Since v satisfies the covering property, we have $v(A) \leqslant v(a) + v(b)$. Now, it is easy to see that v is submodular. □

Theorem 16 and Lemma 17 give:

Corollary 18. *Market* 2 *is an SUA market for the case of two source-sink pairs.*

Hence, Algorithm 9 computes equilibrium for this case in strongly polynomial time. We give below are more efficient algorithm by invoking Theorem 16. The following lemma puts a useful restriction.

Lemma 19. *At most one of the following conditions can be violated*:

- $m_1/(m_1 + m_2) \leqslant F_1/F$,
- $m_2/(m_1 + m_2) \leqslant F_2/F$.

Proof. Suppose both conditions are violated. Add the two inequalities to get $F > F_1 + F_2$. Clearly, the union of a minimum $s_1 - t_1$ cut and a minimum $s_2 - t_2$ cut is a cut separating both pairs of terminals. Hence, $F \leqslant F_1 + F_2$. The contradiction establishes the lemma. □

By Lemma 19 there are three cases; our algorithm needs to distinguish these cases.

Case 1. $m_1/(m_1 + m_2) \leqslant F_1/F$ and $m_2/(m_1 + m_2) \leqslant F_2/F$.

In this case only the edges of S will have positive prices. Price each unit of capacity across cut S at $(m_1 + m_2)/F$. The flow required from s_i to t_i is $(m_i F)/(m_1 + m_2)$ which is at most F_1, for $i = 1, 2$. Furthermore, the sum of flows required is exactly F. By Theorem 16 such a flow is feasible. It can be found with one max-flow computation (by introducing a new source and sink together with appropriate capacity edges to the terminals). Clearly this flow must saturate every edge crossing S. Since the capacity of S equals F, each flow path must cross S exactly once. Hence all KKT conditions are satisfied.

Case 2. $m_1/(m_1+m_2) > F_1/F$ and $m_2/(m_1+m_2) \leqslant F_2/F$.

In this case we will route F_1 flow from s_1 to t_1 and $F-F_1$ flow from s_2 to t_2. By Theorem 16 this flow is feasible. It must saturate edges of S_1 and S. Now, price the edges S at rate $m_2/(F-F_1)$ and the edges of S_1 at rate $(m_1/F_1)-(m_2/(F-F_1))$; by the assumption of this case, the latter is positive. Edges contained in both S and S_1 will get the sum of the two prices i.e., m_1/F_1. Since the capacity of edges in S_1 is F_1, no s_2-t_2 flow can cross this cut. Each flow path must cross S exactly once and each flow path flow from s_1 to t_1 must cross S_1 exactly once. Hence all KKT conditions are satisfied.

Case 3. case $m_1/(m_1+m_2) \leqslant F_1/F$ and $m_2/(m_1+m_2) > F_2/F$.

This case is analogous to Case 2.

Example 20. We give an example to show that Market 1 is not a UUA market for the case of two source-sink pairs. Consider a directed graph with two source-sink pairs, s_1, t_1 and s_2, t_2, and additional vertices a_i, b_i, for $1 \leqslant i \leqslant n$. The graph has the following edges, all of unit capacity: for $1 \leqslant i \leqslant n$ the edges $(s_2, a_i), (a_i, b_i), (b_i, t_2)$, for $1 \leqslant i \leqslant n-1$ the edges (b_i, a_{i+1}), and the edges $(s_1, a_1), (b_n, t_1)$.

Let f_1 (f_2) be the flow sent from s_1 to t_1 (s_2 to t_2). The packing constraint in the Eisenberg–Gale-type convex program is:

$$nf_1 + f_2 \leqslant n,$$

hence showing that this is not a UUA market.

Example 21. Using an example appearing in Hu (1963), we show that Market 2 is not a UUA market for the case of three source-sink pairs. Consider an undirected graph with three source-sink pairs, s_1-t_1, s_2-t_2 and s_3-t_3, and the edges

$$(s_1, s_2),\ (s_1, s_3),\ (s_1, t_2),\ (s_2, t_1),\ (s_2, t_3),\ (t_1, t_2),\ (t_1, s_3),\ (t_2, s_3),$$

all of unit capacity. Let f_1, f_2, f_3 be the flows sent on the three source-sink pairs. Capacity constraints imply:

$$f_1 \leqslant 3, \qquad f_2 \leqslant 3, \qquad f_3 \leqslant 2,$$
$$f_1+f_2 \leqslant 4, \qquad f_1+f_3 \leqslant 3, \qquad f_2+f_3 \leqslant 3,$$
$$f_1+f_2+f_3 \leqslant 4.$$

On the other hand, the flow $f_1 = 1,\ f_2 = 1,\ f_3 = 2$, which satisfies all these constraints, cannot be routed.

9. The spanning tree market in undirected graphs

Using Theorem 22 we show that Market 4 is an SUA market. We also use this theorem to give a strongly polynomial algorithm for this market that is more efficient than Algorithm 9. Assume that agent s has money m_s, $s \in A$, and m is the total money with the agents. Observe that this market is quite different from the others in that each agent desires the same objects, namely spanning trees.

For a partition $V_1, \ldots, V_k$, $k \geqslant 2$, of the vertices, let C be the total capacity of edges whose end points are in different parts. Let us define the edge-tenacity of this partition to be $C/(k-1)$, and let us define the *edge-tenacity* of G to be the minimum edge-tenacity over all partitions. By a *fractional packing of spanning trees* in G we mean choosing each spanning tree T of G to an extent of x_T, which is allowed to be any non-negative number, such that the total extent to which each edge of G is used is no more than its capacity. Such a packing is *maximum* if $\sum_T x_T$ is maximum possible. We will use the following max-min relation.

Theorem 22. *(See Nash-Williams (1961), and Tutte (1961).) The maximum fractional packing of spanning trees in an undirected graph is exactly equal to its edge-tenacity.*

Let t be the edge-tenacity of the given graph and f_s be the number of spanning trees allocated to agent s, $s \in A$. The packing constraints on these variables are

$$\sum_{i \in S} f_i \leqslant t,$$

for any $S \subseteq A$. This is clearly an SUA market.

Find the partition with minimum edge-tenacity using algorithms of Barahona (1995), Cunningham (1985). Let C be the capacity of all the crossing edges of the partition and k the number of parts. Compute the maximum fractional packing of spanning trees. This fractional packing will saturate all the crossing edges of the partition. Also note that each spanning tree in the packing will be using exactly $k-1$ crossing edges of the partition. Price the crossing edges of the partition at rate m/C and assign m_s/m trees to agent s. It is easy to see that all KKT conditions are satisfied.

10. The structure of Market 3

We will first use Edmonds' theorem, Theorem 23, to show that Market 3 is a UUA market. In addition, we will show that it is an SUA market iff $k \leqslant 2$, where k is the number of sources.

Theorem 23. *(See Edmonds (1967).) Let $G = (V, E)$ be a directed graph with edge capacities specified and source $s \in V$. The maximum number of branchings rooted out of s that can be packed in G equals $\min_{v \in V} c(v)$, where $c(v)$ is the capacity of a minimum $s - v$ cut.*

Let $\mathcal{M}$ be a market defined via directed graph $G = (V, E)$ with edge capacities specified. Assume that the set of sources of this market is $A = \{s_1, \ldots, s_k\}$, $S \subseteq V$. Assume that the money of source s_i is m_i. Define function $v : A \to \mathbf{R}_+$ as follows. For $S \subseteq A$, let $v(S)$ be the capacity of the minimum cut separating a vertex in $V - S$ from S.

Theorem 24. *We can pack in G f_i branchings rooted at s_i, for $1 \leqslant i \leqslant k$, simultaneously iff*

$$\forall S \subseteq A: \quad \sum_{i \in S} f_i \leqslant v(S).$$

Proof. The forward implication is obvious. For the reverse implication, add to G a new source vertex s with edges (s, s_i) of capacity f_i, for $1 \leqslant i \leqslant k$, to obtain graph G'.

Let $(S, V - S)$ be a min-cut in G', as required in Theorem 23, with $s \in S$. Let T be the set of sources in S, i.e., $T = S \cap A$. Now,

$$\operatorname{cap}(S, V - S) \geqslant \sum_{i \in A - T} f_i + v(T) \geqslant f_1 + \cdots + f_k,$$

where the last inequality follows from the fact that $f_1, \ldots, f_k$ satisfy the constraints specified in the statement of the lemma. Now, by Theorem 23, $f_1 + \cdots + f_k$ branchings, rooted at s, can be packed in G'. Of these, f_i branchings must use the edge (s, s_i) for $1 \leqslant i \leqslant k$. The theorem follows. □

Corollary 25. *Market 3 is a UUA market.*

This is additionally an SUA market for the case of at most two sources—this follows by arguments given in Section 8 and Lemma 17. For three or more sources, this market is not an SUA market as shown below.

Example 26. Consider a directed graph on vertex set $A = \{a, b, c\}$ and four edges (a, b), (b, a), (c, b), (b, c), with capacities 10, 1, 10, 2, respectively. Assume that a, b and c are all sources. Let function $v : 2^A \to \mathbf{R}_+$ specify the maximum number of branchings that can be rooted at subsets of A. Now, the capacity of (b, a) gives the constraint $v(\{b, c\}) = 1$ and the capacity of (b, c) gives $v(\{a, c\}) = 2$. It is easy to see that $v(\{c\}) = 1$ and $v(\{a, b, c\}) = 3$. Therefore, v is not submodular.

Finally, we give more efficient algorithms for the case of at most two sources. If the market has only one source, equilibrium follows in a straightforward manner from Edmonds' theorem. Let v be a vertex attaining the minimum in Theorem 23, and C be the set of edges in a min $s - v$ cut. Price the edges of C at rate $m/\operatorname{cap}(C)$ where m is the money with the source.

Next assume the market has two sources s_1, s_2 with money m_1, m_2, respectively. Let F_1 and F_2 be capacities of a minimum $s_1 - s_2$ and $s_2 - s_1$ cut, respectively. Let F be $\min_{v \in V - \{s_1, s_2\}} f'(v)$, where $f'(v)$ is the capacity of a minimum cut separating v from s_1 and s_2. The next corollary follows from Theorem 24 and is analogous to Theorem 16.

Corollary 27. *The maximum number of branchings, rooted at s_1 and s_2, that can be packed in G is exactly* $\min\{F_1 + F_2, F\}$.

Let f_1 and f_2 be two non-negative real numbers such that $f_1 \leqslant F_1$, $f_2 \leqslant F_2$ and $f_1 + f_2 \leqslant F$. Then there exists a packing of branchings in G with f_1 of them rooted at s_1 and f_2 of them rooted at s_2.

We show how to use Corollary 27 to compute equilibrium prices and allocations. Our algorithm needs to distinguish four cases:

Case 0. $F_1 + F_2 < F$.

In this case, simultaneously, F_1 branchings can be rooted at s_1 and F_2 branchings can be rooted at s_2, and clearly, neither root can support more branchings individually. Consider such a packing of branchings. Let c_1 be the set of edges in a minimum $s_1 - s_2$ cut (which is of capacity F_1) and c_2 be the set of edges in a minimum $s_2 - s_1$ cut. Clearly, this packing must saturate both these cuts and therefore they must be disjoint. Now price edges of c_1 at rate $m_1/\operatorname{cap}(c_1)$ and edges of c_2 at rate $m_2/\operatorname{cap}(c_2)$. Clearly, all KKT conditions are satisfied.

Next assume that $F \leqslant F_1 + F_2$. Now, the situation is completely analogous to that in Section 8, and the three cases given there yield equilibrium prices and allocations.

11. Efficiency, fairness, and competition monotonicity

Let $\mathcal{M}(v)$ be a UUA market and I be an instance of this market. Under I let the money of agent i be m_i. Let $M(I)$ denote the set of agents having positive money in instance I, i.e., $M(I) = \{i \in A \mid m_i > 0\}$. Assume $|M(I)| = n$ and that $M(I) = \{1, \ldots, n\}$.

Since v is a covering function, by the Bondareva–Shapley Theorem (Bondareva, 1963; Shapley, 1967), there is a utility vector that is feasible for v and it makes set $M(I)$ tight, i.e., the total utility derived by agents having money is $v(M(I))$. Denote by $u(I) = u_1 + \cdots + u_n$ the total utility derived by agents in $M(I)$ at the equilibrium utility vector $\boldsymbol{u}$. We will define the *efficiency of this market*

$$\text{efficiency}\big(\mathcal{M}(v)\big) = \min_I \frac{u(I)}{v(M(I))},$$

where the minimum is over all instances of $\mathcal{M}(v)$.

Let $\boldsymbol{q}$ and $\boldsymbol{r}$ be n-dimensional vectors with non-negative coordinates. We will denote by $\boldsymbol{q}_{DEC}$ the vector obtained by sorting the components of $\boldsymbol{q}$ in decreasing order, and will say that $\boldsymbol{q}$ *min-max dominates* $\boldsymbol{r}$ if $\boldsymbol{q}_{DEC}$ is lexicographically smaller than $\boldsymbol{r}_{DEC}$, i.e., if there is an i such that $\boldsymbol{q}_{DEC}(i) < \boldsymbol{r}_{DEC}(i)$ and $\boldsymbol{q}_{DEC}(j) = \boldsymbol{r}_{DEC}(j)$ for $j < i$. Clearly, $\boldsymbol{q}_{DEC} = \boldsymbol{r}_{DEC}$ may hold even though $\boldsymbol{q} \neq \boldsymbol{r}$.

Similarly, we will denote by $\boldsymbol{q}_{INC}$ the vector obtained by sorting the components of $\boldsymbol{q}$ in increasing order, and will say that $\boldsymbol{q}$ *max-min dominates* $\boldsymbol{r}$ if $\boldsymbol{q}_{INC}$ is lexicographically larger than $\boldsymbol{r}_{INC}$, i.e., if there is an i such that $\boldsymbol{q}_{INC}(i) > \boldsymbol{r}_{INC}(i)$ and $\boldsymbol{q}_{INC}(j) = \boldsymbol{r}_{INC}(j)$ for $j < i$.

We will define core(I) to be the set of functions $f : A \to \mathbf{R}_+$ such that f is feasible for v and f makes $M(I)$ tight. $f \in \text{core}(I)$ will be said to be *min-max fair* if $(f_1/m_1, \ldots, f_n/m_n)$ min-max dominates $(g_1/m_1, \ldots, g_n/m_n)$ for all functions $g \in \text{core}(I)$. Observe that here we are comparing the bang-per-buck of agents. Similarly, f will be said to be *max-min fair* if $(f_1/m_1, \ldots, f_n/m_n)$ max-min dominates $(g_1/m_1, \ldots, g_n/m_n)$ for all functions $g \in \text{core}(I)$.

Next we define the notion of *competition monotonicity* for a market $\mathcal{M}$. Let I be an instance of $\mathcal{M}$ with agents i, j having money m_i and m_j, respectively. Obtain instance I' by increasing the money of agent i from m_i to m'_i, $m'_i > m_i$, and keeping the money of the rest of the agents unchanged. Let $\boldsymbol{u}$ and $\boldsymbol{u}'$ be the equilibrium utility vectors for the two instances. We will say that market $\mathcal{M}$ possesses the property of competition monotonicity if for any two such instances, I and I', $\boldsymbol{u}'(j) \leqslant \boldsymbol{u}(j)$, i.e., on increasing the money of i the utility of agent j cannot increase.

The following fact is well known. Let $\mathcal{M}$ be a Fisher market satisfying weak gross substitutability. Suppose at prices $\boldsymbol{p}$ the total demand for each good is at least as large as the supply. Then, the equilibrium price vector must dominate $\boldsymbol{p}$, i.e., the equilibrium price of each good j must be at least p_j. We use this to prove:

Theorem 28. *Let $\mathcal{M}$ be a Fisher market satisfying weak gross substitutability. Then $\mathcal{M}$ satisfies competition monotonicity.*

Proof. Let I be an instance of $\mathcal{M}$ with each agent j having money m_j. Let $\boldsymbol{p}$ and $\boldsymbol{u}$ be equilibrium prices and utilities. Now, assume the money of agent i is increased to m'_i, keeping the money of the rest of the agents unchanged. At prices $\boldsymbol{p}$, the demand of agent i for each good can only go up and the demands of the rest of the agents remains unchanged. Therefore, by the fact stated above, the new equilibrium prices will dominate $\boldsymbol{p}$. Therefore the allocations, and hence utilities, for each of the other agents cannot increase. □

12. Efficiency of SUA markets

In this section we will prove the following theorem.

Theorem 29. *A uniform utility allocation market has efficiency* 1 *iff it is a submodular utility allocation market.*

One direction of this theorem, that an SUA market has efficiency 1, follows directly from the fact that the equilibrium found by Algorithm 9 makes set $A = M(I)$ tight.

For the other direction, consider a UUA market $\mathcal{M}(v)$ which is defined via function $v : A \to \mathbf{R}_+$ that satisfies the covering property but is not submodular. Let S be a smallest sized set for which submodularity is violated, i.e., there exist $i, j \notin S$ such that:

$$v\big(S \cup \{i\}\big) + v\big(S \cup \{j\}\big) < v(S) + v\big(S \cup \{i, j\}\big).$$

Let

$$v\big(S \cup \{i, j\}\big) = v\big(S \cup \{i\}\big) + v\big(S \cup \{j\}\big) - v(S) + \delta,$$

where $\delta > 0$. Since v satisfies the covering property it is easy to see that $S \neq \emptyset$.

We wish to construct an instance I of this market such that $M(I) = S \cup \{i, j\}$, yet the equilibrium utility vector does not tighten the set $S \cup \{i, j\}$. One way to ensure the latter condition is to ensure that at equilibrium, set S is tight—since the equilibrium is feasible for $S \cup \{i\}$ and $S \cup \{j\}$, $S \cup \{i, j\}$ cannot be tight. However, it is easy to construct function v such that S is tight only when i and j do not have any money, e.g., let $v(S \cup \{i\}) = v(S \cup \{j\}) = v(S)$ and $v(S \cup \{i, j\}) = v(S) + 1$.

We will circumvent this difficulty essentially by showing an instance for which S is almost tight. Then $v(S \cup \{i, j\})$ will not be tight as shown in Lemma 30. Suppose all agents in S have unit money and i and j have no money. Since v is submodular over S, S will be tight at equilibrium. Now, if we could show continuity of the total equilibrium utility of S as the money of i and j is in the neighborhood of zero, we would be done.

Lemma 30. *Let g be a feasible utility allocation for v such that $g(S) > v(S) - \delta$. Then g cannot tighten $S \cup \{i, j\}$.*

Proof. Since g is a modular function,

$$\begin{aligned} g\big(S \cup \{i, j\}\big) &= g\big(S \cup \{i\}\big) + g\big(S \cup \{j\}\big) - g(S) \\ &< v\big(S \cup \{i\}\big) + v\big(S \cup \{j\}\big) - v(S) + \delta = v\big(S \cup \{i, j\}\big). \end{aligned}$$

The lemma follows. □

Let L be the maximum of $v(T)$ over all $T \subseteq S \cup \{i, j\}$. Choose $\epsilon = \delta/2L$. Under instance I, i and j will have ϵ money each and each agent in S will have unit money. Let g be the equilibrium utility vector; in Lemma 32 we will show that $g(S) > v(S) - \delta$.

Lemma 31. *There is a feasible vector, h, for v such that $h(i) = h(j) = 0$, and for $l \in S$, $h(l) \geqslant g(l)$ and $\sum_{l \in S} h(l) = v(S)$.*

Proof. Define function $v' : S \to \mathbf{R}_+$ as follows: for $T \subseteq S$, $v'(T) = v(T) - \sum_{l \in T} g(l)$. Recall that v is submodular over S. Since v' is the difference of a submodular and a modular function, it is also submodular. Using the algorithm in Jain and Vazirani (2002) we can find a vector h' which is feasible for v' and makes $v'(S)$ tight. Now define $h(i) = h(j) = 0$ and $h(l) = g(l) + h'(l)$ for all $l \in S$. □

Lemma 32. $g(S) > v(S) - \delta$.

Proof. Since g and h are feasible for v, any convex combination of g and h is also feasible. Take the convex combination $(1 - \lambda)g + \lambda h$, where $\lambda \in [0, 1]$. At $\lambda = 0$ this combination is g and $\lambda = 1$ it is h. Since market equilibrium maximizes $\sum_{i \in A} m_i \log u_i$ and g gives equilibrium utilities, the function

$$\epsilon \log\big((1 - \lambda)g(i)\big) + \epsilon \log\big((1 - \lambda)g(j)\big) + \sum_{l \in S} \log\big((1 - \lambda)g(l) + \lambda h(l)\big)$$

is maximized at $\lambda = 0$. Therefore, the derivative of this expression with respect to λ at $\lambda = 0$ is non-negative, i.e.,

$$-\epsilon - \epsilon + \sum_{l \in S} \frac{h(l) - g(l)}{g(l)} \leqslant 0.$$

This implies

$$\sum_{l \in S} \frac{h(l) - g(l)}{g(l)} \leqslant 2\epsilon.$$

Since $h(l) \geqslant g(l)$, all the terms on the left-hand side are non-negative. This implies

$$\forall l \in S: \quad h(l) - g(l) \leqslant 2\epsilon g(l).$$

Summing over all $l \in S$ gives

$$v(S) - \sum_{l \in S} g(l) \leqslant 2\epsilon \sum_{l \in S} g(l) < 2\epsilon v(S) \leqslant 2\epsilon L = \delta.$$

Hence, $g(S) > v(S) - \delta$. □

Hence for instance I of market $\mathcal{M}(v)$, equilibrium utilities do not make the set $M(I)$ tight.

13. Fairness of equilibria for SUA markets

In this section we will prove the following theorem.

Theorem 33. *The equilibrium for a submodular utility allocation market is max-min fair and min-max fair.*

Proof. Let I be an instance of an SUA market $\mathcal{M}(v)$. Let $M(I) = A$ and $|A| = n$. Let $u \in \mathrm{core}(I)$ be an allocation of equilibrium utilities as computed by Algorithm 9. Let

$$A = A_1 \supset A_2 \supset \cdots \supset A_k \supset A_{k+1} = \emptyset$$

be the sets that go tight in the run of the algorithm.

We will first prove that u is min-max fair. Let $u'(i) = u(i)/m_i$ for $i \in A$; this function gives equilibrium bang-per-buck of agents. Let $g \in \mathrm{core}(I)$ and $g'(i) = g(i)/m_i$ for $i \in A$. Suppose g' min-max dominates u'. Then we will show that $u' = g'$.

We will show by induction on i that for all agents $j \in A_i - A_{i+1}$, $u'(j) = g'(j)$; clearly, all these agents have the same rates and hence the same bang-per-buck in the equilibrium allocation. Now,

$$\sum_{i \in A_2} g(i) \leqslant \sum_{i \in A_2} u(i) = v(A_2).$$

Since g makes A tight,

$$\sum_{i \in A_1 - A_2} g(i) \geqslant \sum_{i \in A_1 - A_2} u(i).$$

If this inequality is strict, $\exists i \in A_1 - A_2$ such that $g(i) > u(i)$. Since agents $i \in A_1 - A_2$ have the highest equilibrium bang-per-buck, u' min-max dominates g', leading to a contradiction. Therefore, this inequality must hold with equality.

If for some user $i \in A_1 - A_2$, $g(i) < u(i)$, then for some other user $j \in A_1 - A_2$, $g(j) > u(j)$, and again u' min-max dominates g', leading to a contradiction. Therefore,

$$\forall i \in A_1 - A_2, \quad g(i) = u(i).$$

The proof of the induction step is along the same lines.

Hence we get that u is min-max fair. The proof further shows that u is the unique min-max fair allocation. In Jain and Vazirani (2002) we showed that for a submodular function v the min-max fair allocation is also the unique max-min fair allocation. This establishes the theorem. □

It is straightforward to construct examples of markets in (UUA–SUA) that are neither max-min nor min-max fair. We next give an example of a market in (UUA–SUA) which is max-min and min-max fair, and we leave the open problem of characterizing the class of max-min and/or min-max fair markets within UUA.

Example 34. Consider a UUA market on a set A of 3 agents which is defined via function v: v has value 3 on any singleton, value 4 on any subset of A of cardinality 2 and value 6 on A. Clearly, v is not submodular. Now, it is easy to see that the core consists of only one function—it assigns 2 to each agent. Therefore, the equilibrium for any instance of this market must be max-min and min-max fair.

14. Competition monotonicity for SUA markets

Theorem 35. *Submodular utility allocation markets satisfy competition monotonicity.*

Proof. We will use the fact that for an SUA market, the equilibrium utility vector is max-min fair, as established in Theorem 33, and the following result from Jain and Vazirani (2002) which we are stating in a language appropriate to this paper.

Let $v : A \to \mathbf{R}_+$ be a submodular function, $\mathcal{M}(v)$ be an SUA market and I be an instance of it; assume w.l.o.g. that $M(I) = A$. Let m_i be the money of agent $i \in A$ under instance I. For $S \subseteq A$, let $I(S)$ denote the instance in which agent $i \in S$ has money m_i and agents in $A - S$ have no money. Let g_S denote the function assigning equilibrium utilities to agents in S under instance $I(S)$. The family of functions g_S, $S \subseteq A$, is said to be *cross-monotonic* if $\forall i \in S \subset T \subseteq A$, $g_S(i) \geqslant g_T(i)$.

By Theorem 33, $\forall S \subseteq A$, g_S is max-min fair. Now, by Jain and Vazirani (2002), g is cross-monotonic (in the language of Jain and Vazirani (2002), the equalizing functions we are using are $f_i(x) = xm_i$, for $1 \leqslant i \leqslant n$).

Let $v : A \to \mathbf{R}_+$ be a submodular function, $\mathcal{M}(v)$ be an SUA market and I be an instance of it; assume w.l.o.g. that $M(I) = A$. Let m_i be the money of agent $i \in A$ under instance I. Obtain instance I' by increasing the money of agent i by ϵ and keeping the money of all other agents the same. Let $\boldsymbol{u}$ and $\boldsymbol{u}'$ be equilibrium utility vectors for instances I and I'. We need to show that for any agent $j \neq i$, $u_j \geqslant u'_j$.

Let $A' = A \cup \{i'\}$, where i' is a new agent. Define $v' : A' \to \mathbf{R}_+$ as follows. For $S \subseteq A$, $v'(S) = v(S)$ and $v'(S \cup \{i'\}) = v(S \cup \{i\})$. Clearly, v' is also submodular. Consider instance J of market $\mathcal{M}(v')$ in which the money of all agents $i \in A$ is the same as in instance I, and the money of i' is ϵ. By the basic property of Eisenberg–Gale-type programs (see Section 2), for $j \in A$, $j \neq i$, the utility of j is the same under instance J of market $\mathcal{M}(v')$ and instance I of market $\mathcal{M}(v)$.

Let g_S, $S \subseteq A'$, be the family of functions giving equilibrium utilities under instance $J(S)$ of market $\mathcal{M}(v')$. As stated above, this family of functions is cross-monotonic. Therefore, for $j \in A$, $j \neq i$, $g_A(j) \geqslant g_{A'}(j)$. But $g_A(j) = u_j$ and $g_{A'}(j) = u'_j$. Therefore, $u_j \geqslant u'_j$. □

Example 36. We give an example of a UUA market, which is not an SUA market, for which competition monotonicity does not hold. The market consists of three agents i, j, and k. Define function v as follows: $v(i) = v(j) = v(k) = 2$, $v(i, j) = v(j, k) = v(k, i) = 2$ and $v(i, j, k) = 3$. Let i, j, k have money $1, 2, 5$, respectively. Then one can check that equilibrium utilities are $3/4$, $3/4$, $5/4$. When the money of i is raised to 2, equilibrium utilities are $8/9$, $8/9$, $10/9$. Observe that the utility for j has increased.

We leave the open question of determining whether competition monotonicity characterizes SUA markets within the class of UUA markets.

15. Efficiency of resource allocation markets

We first define the notion of efficiency of a resource allocation market, much the same way it was defined for UUA markets. Let $\mathcal{M}$ denote a resource allocation market. Since the objective function of convex program (2) for $\mathcal{M}$ is strictly concave, one can see that at optimality, the vector $f_1, \ldots, f_n$ is unique. Clearly, this also holds for every equilibrium allocation. For instance I of $\mathcal{M}$ let $f(I)$ denote the total number of objects made by the agents in an equilibrium allocation; clearly, $f(I) = f_1 + \cdots + f_n$, where $f_1, \ldots, f_n$ is the unique vector optimizing convex program (2). Denote by $F(I)$ the maximum number of objects that can be made in a feasible allocation of resources. We will define

$$\text{efficiency}(\mathcal{M}) = \min_{I \in \mathcal{M}} \frac{f(I)}{F(I)}.$$

Each of the resource allocation markets for which we have obtained strongly polynomial algorithms are also SUA markets. Therefore, by Theorem 29 they have efficiency 1. We next give bounds on the efficiency of the remaining markets.

15.1. Multiple source-sink markets for directed and undirected graphs

We first give an example to show that for the case of directed graphs, the efficiency of the corresponding market can be arbitrarily small, even for only two source-sink pairs.

Example 37. Consider the graph given in Example 21. The maximum number of flow paths this graph supports is the n paths from s_2 to t_2. Assume the pair s_1, t_1 has \$$H$, where H is a large number, and the pair s_2, t_2 has \$1. In this case the equilibrium is $1 - \epsilon$ path from s_1 to t_1 and ϵn paths from s_2 to t_2, where ϵ approaches 0 as H approaches infinity. Therefore, the efficiency of this market approaches $1/n$, and hence can be made arbitrarily small.

In contrast, for the case of undirected graphs, the efficiency of this market is a function of the number of source-sink pairs. We will show below that it is at least $1/(2k-1)$ where k is the number of source-sink pairs. The best upper bounding example we have is $1/(k-1)$.

Consider a valid flow that sends f_i flow for the ith source-sink pair, for $1 \leqslant i \leqslant k$. We will say that this flow is *Pareto optimal* if none of the f_i's can be increased, i.e., this a maximal, though not necessarily maximum, flow.

Lemma 38. *If $f_1, f_2, \ldots, f_k$ be a Pareto optimal assignment of flows, then $f_1 + f_2 + \cdots + f_k \geqslant \frac{1}{2k-1} OPT$, where OPT is the maximum multicommodity flow that can be sent in G.*

Proof. Consider a realization of flow vector $f_1, f_2, \ldots, f_k$ in such a way that it minimizes the number of tight edges. Let G' be the graph obtained after contracting all edges that are not saturated. Clearly, for each i, s_i and t_i will not collapse into a single node, since otherwise Pareto optimality is violated.

It is easy to see that $f_1, f_2, \ldots, f_k$ is a Pareto optimal flow vector for G' as well. Suppose, for contradiction, $f_1 + \epsilon$, $f_2, \ldots, f_k$ is a feasible vector for G' for $\epsilon > 0$. Among edges of G that are not saturated, let ϵ' be the smallest difference between capacity and flow; clearly, $\epsilon' > 0$. Then, it is easy to see that $f_1 + \min\{\epsilon, \epsilon'\}, f_2, \ldots, f_k$ is a feasible vector for G, leading to a contradiction.

Let OPT' be the maximum multicommodity flow G'. Clearly OPT' is at least OPT, because every feasible solution for G is feasible for G' as well. Hence, it suffices to prove the lemma for G'.

Consider all nodes of G' that have at least one s_i collapsed into it, and consider the set of edges incident at these nodes. Say that these edges are *marked*. Let C be the total capacity of these edges in G'. Clearly C is an upper bound on OPT', because any flow must use at least one such edge. Now we will prove that $f_1 + f_2 + \cdots + f_k \geqslant \frac{1}{2k-1}C$.

Consider the flow supporting vector $f_1, f_2, \ldots, f_k$ on G'. It is again easy to see that this flow cannot have any cycles because otherwise we can cancel a cycle, leading to a decrease in the number of saturated edges. This acyclicity implies that any flow path uses at most $2k-1$ marked edges. Since $f_1, f_2, \ldots, f_k$ tightens all the marked edges we get that $f_1 + f_2 + \cdots + f_k \geqslant \frac{1}{2k-1}C$. □

Since any market equilibrium must be Pareto optimal, we get.

Theorem 39. *The efficiency of multiple source-sink market for undirected graphs is at least* $1/(2k-1)$ *where k is the number of source-sink pairs.*

Example 40. We give an example showing an upper bound of essentially $1/(k-1)$ on the efficiency of this market. The graph is a path of length $k-1$ on vertices $v_1, \ldots, v_k$, consisting of unit capacity edges. There are k source-sink pairs: (v_i, v_{i+1}), for $1 \leqslant i \leqslant k-1$, and (v_1, v_k). The maximum multicommodity flow has value $k-1$ and sends one unit of flow on the first $k-1$ source-sink pairs. Next, suppose each of the first $k-1$ source-sink pairs have \$1 and (v_1, v_k) has \$$H$, where H is a large number. Now, the equilibrium will send $1-\epsilon$ flow of the last commodity and ϵ flow of the remaining $k-1$ commodities, where ϵ approaches zero as H approaches infinity.

15.2. The branching market with three or more sources

Let $A \subseteq V$ be the set of sources, with $|A| \geqslant 3$. For $S \subset V$, let $\mathrm{in}(S)$ denote the capacity of edges incoming into set S, and out(S) denote the capacity of edges outgoing from set S. Let $B_1 = \min\{\mathrm{in}(S)\}$, where the minimum is over all sets of vertices such that $S \cap A = \emptyset$, $S \neq \emptyset$. Let $B_2 = \min\{\mathrm{in}(S_1) + \mathrm{in}(S_2)\}$, where the minimum is over all disjoint sets of vertices S_1 and S_2. Let $B = \min\{B_1, B_2\}$.

We will first upper bound the optimum number of branchings that can be packed in G. Note that such a step is critical in developing an approximation algorithm for an **NP**-hard maximization problem. Our reason is different; in fact in our case, the optimum number can be computed in polynomial time via a linear program.

Lemma 41. *B is an upper bound on the number of branchings, rooted at vertices of A, that can be packed in G.*

Proof. Suppose the minimum is attained by B_1 via set S. Since $S \cap A = \emptyset$ and $S \neq \emptyset$ any branching rooted at a vertex of A must use an edge of $\mathrm{in}(S)$ to reach into S, giving the upper bound. Next suppose the minimum is attained by B_2 via disjoint sets S_1 and S_2. Now branchings rooted at $\overline{S_1}$ must enter S_1 using edges of $\mathrm{in}(S_1)$ and those rooted at vertices of $S_1 \cap A$ must enter S_2 using edges of $\mathrm{in}(S_2)$. Hence the bound. □

Let $A = \{s_1, \ldots, s_k\}$. Consider a valid packing of branchings that packs f_i branchings rooted at s_i, for $1 \leqslant i \leqslant k$. We will say that this packing is *Pareto optimal* if none of the f_i's can be increased.

Lemma 42. *For any Pareto optimal packing, $F = f_1 + \cdots + f_k \geqslant B/2$.*

Proof. Obtain graph G' from G by adding new vertex s and edges (s, s_i) of capacity f_i, for $1 \leqslant i \leqslant k$. Consider min $s-v$ cuts in G', for $v \in V$, and pick the minimum capacity cuts among these. Let $(\overline{S}, S)$ be one such cut, $s \in \overline{S}$. By Edmonds' theorem, the maximum number of branching rooted at s that can be packed in G', which is clearly F, is the in-capacity of S in G'. Clearly, in any maximum packing, S must be *tight*, i.e., each branching must enter S exactly once and all of the in-capacity of S must be used by the packing. If there are two minimal such sets, say S_1 and S_2, then it is easy to see that they cannot cross, and hence must be disjoint.

Case 1. There exist two minimal such sets, say S_1 and S_2.

Clearly, S_1 and S_2 must be tight in the packing being considered in G. Assume w.l.o.g., that $\mathrm{in}(S_1) \geqslant \mathrm{in}(S_2)$. By Lemma 41, $\mathrm{in}(S_1) \geqslant B/2$. Note that the in-capacities of these sets are being considered in graph G, i.e., without the edges (s, s_i), $1 \leqslant i \leqslant k$. Now, since S_1 is tight, $F \geqslant \mathrm{in}(S_1) \geqslant B/2$.

Case 2. There exists only one minimal such set, S, and $S \cap A = \emptyset$.

Since S is tight, and it does not contain any sources, $F = \mathrm{in}(S) \geqslant B$.

Case 3. There exists only one minimal such set, S, and $s_i \in (S \cap A)$.

In this case, all minimum $s - v$ cuts in G' must contain S. Therefore, edge (s, s_i) is contained in each of these cuts. Therefore, we can increase the capacity of this edge to $f_i + \epsilon$ for a small $\epsilon > 0$ without introducing a new min-cut. Now, by Edmonds' theorem, the maximum number of branchings rooted at s that can be packed in G' is the in-capacity of S in G' and is $F + \epsilon$, with $f_i + \epsilon$ of them using the edge (s, s_i). This contradicts Pareto optimality of the packing. Hence this case cannot arise. □

Theorem 43. *The efficiency of the branching market is* $\geqslant 1/2$.

Proof. An equilibrium packing must be an optimal solution to convex program (2). Since the objective function of this program is a strictly monotone function of the packing $f_1, \ldots, f_k$, an equilibrium packing must be Pareto optimal. The theorem follows from Lemmas 41 and 42. □

Remark 44. Observe that for the case of two sources, by Theorem 24 and Corollary 25, any Pareto optimal solution is exactly B, hence leading to efficiency 1.

Example 45. We next give an example to show that the efficiency of this market can be arbitrarily close to 1/2. Consider a directed graph on three vertices, a, b, c and edges (a, b), (c, b) of capacity infinity and (b, a), (b, c) of capacity 1. All three of these vertices are sources, with a and c having \$1 each and b having \$$H$, where H is a large number. The optimum solution is to build one branching each at a and c; however, the market equilibrium builds $1 - \epsilon$ branching at b and ϵ branching each at a and c. Hence, the efficiency of this market approaches 1/2 as H approaches infinity.

16. Rationality of equilibria

Clearly, each of the markets for which we obtain a strongly polynomial algorithm, there is always a rational equilibrium allocation and prices, provided the input is all rational. For most of the remaining markets, we provide instances that have only irrational solutions.

Example 46. For Markets 1 and 2, if there are two sources and three or more sinks, an irrationality example was given in Garg et al. (2005). We state it again for sake of completeness. Consider a graph on three nodes $\{a, b, c\}$ and two edges (a, b), (b, c). Let the capacity of (a, b) be one and the capacity of (b, c) be two. The source sink pairs are: $a - b$, $a - c$ and $b - c$. Each source-sink pair has \$1. Now, the equilibrium price for (a, b) is $\sqrt{3}$ and for (b, c) it is $\frac{\sqrt{3}}{1+\sqrt{3}}$.

A generalization of the resource allocation model of Kelly, defined in Section 3, was suggested by Kelly and Vazirani (2002). In this generalization, agent a_i derives different utilities on making an object from each of the sets of resources $S_{i1}, S_{i2}, \ldots, S_{ik_i}$. Consider this generalization of the single-source multiple-sink market. The following example admits only irrational prices.

Example 47. The example is essentially the same as Example 46, except that there is a parallel edge (a, b) of very large capacity. Assume source-sink pair $a - c$ prefers to get flow using this edge and the edge (b, c). The remaining source-sink pairs desire flows in the original graph. Clearly, the prices given above apply.

We next give an example having only irrational solutions for Market 3 for the case of three sources.

Example 48. The graph is the same as that in Example 26. It has three vertices a, b, and c and four edges (a, b), (b, a), (c, b), (b, c), with capacities 10, 1, 10, 2, respectively. Assume that a, b and c have \$1 each. Clearly the price of incoming edges into b is zero, because these edges can't be saturated. One can check that at equilibrium, the price of (b, a) is $\sqrt{3}$ and the price of (b, c) is $(3 - \sqrt{3})/2$. One can also check that there is no other equilibrium. Note, that this is related to the tight example we gave in Section 10.

16.1. Rationality of equilibria for UUA markets

By Corollary 25 the last example with three sources is a UUA market, thereby showing that there are UUA markets which have only irrational solutions. Algorithm 9 shows that SUA markets always have rational equilibria if all input parameters are rational. This raises the following question: Does rationality of equilibria characterize SUA markets within the class of UUA markets? We show below that the answer to this question is "no" by giving an instance of a UUA market which is not an SUA market and yet it has rational solutions. This raises several interesting questions (see Section 17).

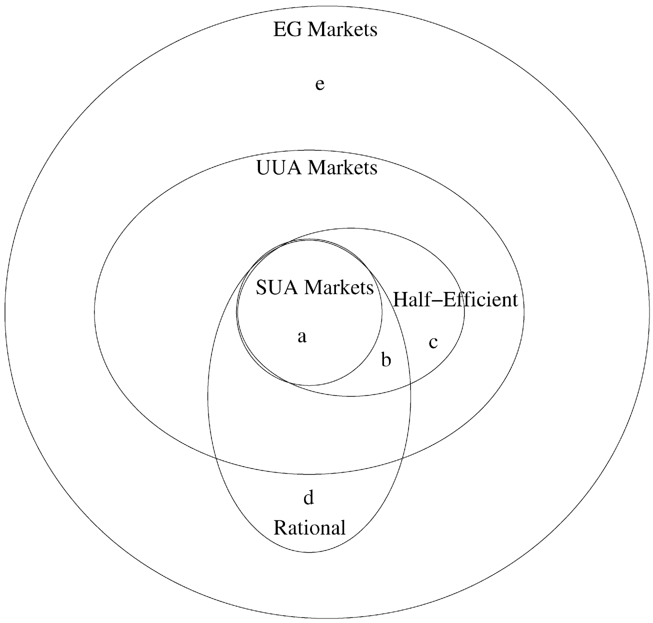

Fig. 2. Venn diagram of markets.

Example 49. Let $A = \{i, j, k\}$ and define v as follows: $v\{i\} = v\{j\} = v\{k\} = v\{i, j\} = v\{j, k\} = v\{i, k\} = 2$ and $v\{i, j, k\} = 3$. Note that v is not submodular. We claim that the UUA market $\mathcal{M}(v)$ has rational equilibria when m_i, m_j, and m_k are rational.

By KKT conditions, only tight sets can have a positive prices. The only way A can be tight is if $u_i = u_j = u_k = 1$. This is so because the sum of any two utilities is bounded by 2, so the third utility must be 1 to ensure that their sum is 3. This is a rational equilibrium.

Next, assume that A is not tight. If singleton set $\{i\}$ is tight then the utilities must be $u_i = 2$, $u_j = u_k = 0$, which is also rational. So we may assume that no singleton set is tight.

Now the only tight sets must have cardinality two. There are three cases. All the three cardinality two sets can be tight only for $u_i = u_j = u_k = 1$, which is rational. If only one of the cardinality two sets is tight, say $\{i, j\}$, then $m_k = 0$ and $p_{\{i,j\}} = (m_i + m_j)/2$. For rational m_i and m_j, this solution is rational.

In the last case two of the cardinality two sets are tight, say $\{i, j\}$ and $\{j, k\}$. Let their prices be $p_{\{i,j\}}$ and $p_{\{j,k\}}$. We get the following two equations at equilibrium:

$$\frac{m_i}{p_{\{i,j\}}} + \frac{m_j}{p_{\{i,j\}} + p_{\{j,k\}}} = 2,$$
$$\frac{m_k}{p_{\{j,k\}}} + \frac{m_j}{p_{\{i,j\}} + p_{\{j,k\}}} = 2.$$

These two equations give:

$$\frac{m_i}{p_{\{i,j\}}} = \frac{m_k}{p_{\{j,k\}}}.$$

Substituting the value of $p_{\{i,j\}}$ in terms of $p_{\{j,k\}}$ in either of the two equations yields rational values for both prices.

Observe that the efficiency of this market is at least 2/3.

17. The picture so far

Fig. 2 summarizes the picture so far. Here "Rational" represents the set of markets having rational solutions and "Half-Efficient" represents the subset of UUA markets having efficiency at least 1/2. The following specific markets have been marked in this figure:

- a: The single-source multiple-sink market.
- b: The market of Example 49.
- c: The market of Example 26.
- d: The linear utilities case of Fisher's model.

e: The Leontief utilities case of Fisher's model, e.g., see Codenotti and Varadarajan (2004). Leontief utilities are defined as follows: Each buyer i is interested in a subset S_i of goods and has coefficient $u_{ij} > 0$, for $j \in S_i$. For allocation $\boldsymbol{x}$, the utility of buyer i is $\min_{j \in S_i} u_{ij} x_{ij}$.

Much work is needed to get a better understanding of these markets. Here are some immediate questions.

1. Is there a characterization of the class of markets admitting rational solutions? Observe that market b in Fig. 2 has rational solutions.
2. Are there strongly polynomial algorithms for UUA markets not contained within SUA markets?
3. Is there a characterization of UUA markets which admit strongly polynomial algorithms?
4. Is there a characterization of EG markets which admit strongly polynomial algorithms?
5. Is there a characterization of UUA markets having efficiency at least 1/2, e.g., by considering a suitable factor 1/2 relaxation of submodularity for defining such markets.
6. Do markets in (UUA–SUA) satisfy a relaxation of max-min or min-max fairness? For relaxed notions of max-min fairness see, for example, Goel et al. (2001).

18. Discussion

As mentioned in the Introduction, finding combinatorial algorithms for solving special classes of linear programs has been extremely valuable in combinatorial optimization. Our own extension of this line of work to convex programs has provided new insights into the efficiency and fairness of markets. It seems fruitful to carry this inquiry further to other classes of nonlinear convex programs.

A natural adaptation of the primal-dual schema, by relaxing complementary slackness conditions, which was first identified and formalized in Williamson et al. (1993), led to approximation algorithms for several fundamental **NP**-hard optimization problems. We believe there should be a corresponding adaptation of this schema for approximately solving nonlinear convex programs.

Following up on the remark made in Kelly and Vazirani (2002) (see Introduction), considering the simplicity of our algorithms and their natural description as ascending price auctions, one wonders if they can provide new insights into the design of congestion control protocols.

Submodular flows (Edmonds, 1970) generalize several optimization problems. It will be interesting to see if the corresponding market, when defined with a single source and multiple sinks, admits a strongly polynomial algorithm. The recent idea of network coding (Ahlswede et al., 2000) in information theory opens up the possibility of using more general structures than branchings for broadcasting: We are given a directed graph $G = (V, E)$ consisting of terminals and Steiner nodes; a subset of the terminals are designated to be sources. An object for source s is a subgraph of G (which, in general, picks edges fractionally) rooted at s and allowing one unit of flow from s to each terminal. If s buys k objects, then it can broadcast at rate k to all terminals, using network coding. What is the structure of this market w.r.t. classes UUA and SUA? Observe that a slight generalization of Kelly's framework is required for formally defining this market since an object picks resources fractionally.

Finally, here are some loose ends in our paper. For Markets 1 and 2, the algorithms given in Sections 4 and 8 use min-max theorems critically; the first uses the max-flow min-cut theorem and the second uses Hu's theorem (Hu, 1963). In all other cases, i.e., the graph is directed or undirected with three or more source-sink pairs (Dahlhaus et al., 1994), or the graph is directed and there are two source-sink pairs (Garg et al., 1994), there is a gap between max-flow and min-cut, and computing the latter is **NP**-hard. For all but the last case, i.e., the graph is directed and there are two source-sink pairs, we have given examples having only irrational solutions. Resolving the remaining case will be interesting.

In Section 13 we showed that all SUA markets are max-min and min-max fair. Whereas it is easy to construct examples of markets in (UUA–SUA) that are neither max-min nor min-max fair, Example 34 gives a market in (UUA–SUA) that is max-min and min-max fair. This leaves the problem of characterizing max-min and/or min-max fair markets in UUA.

In Section 14 we showed that all SUA markets satisfy competition monotonicity. We leave the open problem of determining if this property characterizes SUA markets within the class of UUA markets. For the multiple source-sink market in undirected graphs, there is a gap between the lower bound on efficiency established in Theorem 39 and the upper bounding given in Example 40.

Note added November 2006. For the open problem stated above for Market 1 when there are two source-sink pairs, Chakrabarty et al. (2006) have given a strongly polynomial algorithm. Instead of using a max-min theorem, they use the full power of LP-Duality Theorem. The issue of whether competition monotonicity characterizes SUA markets within the class of UUA markets was settled positively by Chakrabarty and Devanur (2006).

Acknowledgments

We wish to thank Bill Cunningham, Lap Chi Lau and Kunal Talwar for valuable discussions, and Deeparnab Chakrabarty and Lei Wang for pointing out Example 34. Finally, it is a pleasure to submit this paper to *GEB*'s Special Issue commemorating Ehud Kalai's 65th birthday!

References

Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W., 2000. Network information flow. IEEE Transactions Info. Theory 46, 1204–1216.

Barahona, F., 1995. Packing spanning trees. Mathematics Operation Res. 20, 104–115.

Bondareva, O.N., 1963. Some applications of linear programming to cooperative games. Prob. Kibernetiki 10, 119–139.

Brainard, W.C., Scarf, H.E., 2000. How to compute equilibrium prices in 1891. Cowles Foundation Discussion Paper, 1270.

Chakrabarty, D., Devanur, N., 2006. On competitiveness in uniform utility allocation markets. Unpublished manuscript.

Chakrabarty, D., Devanur, N., Vazirani, V., 2006. New results on rationality and strongly polynomial solvability in Eisenberg–Gale markets. In: WINE.

Codenotti, B., Varadarajan, K., 2004. Efficient computation of equilibrium prices for markets with Leontief utilities. In: ICALP.

Cunningham, W.H., 1985. Minimum cuts, modular functions and matroid polyhedra. Networks 15, 205–215.

Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M., 1994. The complexity of multiterminal cuts. SIAM J. Computing 23, 864–894.

Devanur, N., Papadimitriou, C.H., Saberi, A., Vazirani, V.V., 2008. Market equilibrium via a primal-dual-type algorithm. J. ACM 55 (5), article 22.

Edmonds, J., 1967. Optimum branchings. J. Res. Nat. Bureau Standards. Section B 71, 233–240.

Edmonds, J., 1970. Submodular functions, matroids, and certain polyhedra. In: Guy, R., Hanani, H., Sauer, N., Sch/Snheim, J. (Eds.), Combinatorial Structures and their Applications. Gordon and Breach, pp. 69–87.

Eisenberg, E., 1961. Aggregation of utility functions. Manage. Sci. 7, 337–350.

Eisenberg, E., Gale, D., 1959. Consensus of subjective probabilities: The Pari-Mutuel method. The Annals Math. Statist. 30, 165–168.

Fleischer, L., Fujishige, S., Iwata, Satoru, 2001. A strongly polynomial-time algorithm for minimizing submodular functions. J. ACM 48, 761–777.

Floyd, S., Jacobson, V., 1993. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions Networking 1 (1), 397–413.

Garg, N., Vazirani, V.V., Yannakakis, M., 1994. Multiway cuts in directed and node weighted graphs. In: Proc. 21st International Colloquium on Automata, Languages, and Programming. In: Lecture Notes in Computer Sci., vol. 820. Springer-Verlag, Berlin, pp. 487–498.

Garg, D., Jain, K., Talwar, K., Vazirani, V., 2005. A primal-dual algorithm for computing Fisher equilibrium in absence of gross substitutability property. In: WINE.

Goel, A., Meyerson, A., Plotkin, S., 2001. Approximate majorization and fair online load balancing. In: SODA.

Groenevelt, H., 1991. Two algorithms for maximizing a separable concave function over a polymatroid feasible region. Europ. J. Operational Res. 54, 227–236.

Hu, T.C., 1963. Multicommodity network flows. Operations Res. 14, 344–360.

Jacobson, V., 1988. Congestion avoidance and control. In: ACM SIGCOMM, pp. 314–329.

Jain, K., Vazirani, V.V., 2001. Approximation algorithms for the metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. J. ACM 48, 274–296.

Jain, K., Vazirani, V.V., 2002. Equitable cost allocations via primal-dual-type algorithms. In: STOC.

Jain, K., Vazirani, V.V., Ye, Y., 2005. Market equilibrium for homothetic, quasi-concave utilities and economies of scale in production. In: SODA.

Kelly, F.P., 1997. Charging and rate control for elastic traffic. Europ. Transactions Telecommunications 8, 33–37.

Kelly, F.P., Vazirani, V.V., 2002. Rate control as a market equilibrium. Unpublished manuscript. Available at: http://www.cc.gatech.edu/vazirani/KV.pdf.

Kelly, F.P., Maulloo, A.K., Tan, D.K.H., 1998. Rate control in communication networks. J. Operational Res. Society 49, 237–252.

Koutsoupias, E., Papadimitriou, C.H., 1999. Worst case equilibria. In: Proceedings of STACS.

Low, S.H., 2003. A duality model of TCP and queue management algorithms. IEEE/ACM Transactions Networking 11 (4), 525–536 (August).

Low, S., Lapsley, D., 1999. Optimization flow control, 1: Basic algorithm and convergence. IEEE/ACM Transactions Networking 7 (6), 861–874.

Low, S.H., Peterson, Larry, Wang, Limin, 2002. Understanding vegas: A duality model. J. ACM 49 (2), 207–235 (March).

Megiddo, N., 1974. Optimal flows in networks with multiple sources and sinks. Math. Programming 7, 97–107.

Megiddo, N., 1988. A note on the complexity of p-matrix lcp and computing an equilibrium. IBM Research Report 6439. Available at: http://theory.stanford.edu/~megiddo/pdf/plcp.pdf.

Megiddo, N., Papadimitriou, C.H., 1991. On total functions, existence theorems, and computational complexity. Theoretical Computer Sci. 81, 317–324.

Nagano, K., 2007. On convex minimization over base polytopes. In: IPCO.

Nash-Williams, C.S.J.A., 1961. Edge-disjoint spanning trees of finite graphs. J. London Math. Society 36, 445–450.

Schrijver, A., 1986. Theory of Linear and Integer Programming. John Wiley & Sons, New York, NY.

Schrijver, A., 2000. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J. Combinatorial Theory, Ser. B 80, 346–355.

Shapley, L.S., 1967. On balanced sets and cores. Naval Res. Logist. Quart. 14, 453–460.

Tutte, W.T., 1961. On the problem of decomposing a graph into n connected factors. J. London Math. Society 36, 221–230.

Vazirani, V.V., 2007. Combinatorial algorithms for market equilibria. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V. (Eds.), Algorithmic Game Theory. Cambridge University Press, pp. 103–134.

Wang, J., Li, L., Low, S.H., Doyle, J.C., 2005. Cross-layer optimization in TCP/IP networks. IEEE/ACM Transactions Networking 13, 582–595.

Williamson, D.P., Goemans, M.X., Mihail, M., Vazirani, V.V., 1993. A primal-dual approximation algorithm for generalized Steiner network problems. In: Proceedings, 25th Annual ACM Symposium on Theory of Computing.