Rationality and Strongly Polynomial Solvability of Eisenberg-Gale Markets with Two Agents

Deeparnab Chakrabarty * Nikhil R. Devanur † Vijay V. Vazirani ‡

Abstract

Inspired by the convex program of Eisenberg and Gale which captures Fisher markets with linear utilities, Jain and Vazirani [STOC, 2007] introduced the class of Eisenberg-Gale (EG) markets. We study the structure of EG(2) markets, the class of Eisenberg-Gale markets with two agents.

We prove that all markets in this class are rational, that is, they have rational equilibrium, and they admit strongly polynomial time algorithms whenever the polytope containing the set of feasible utilities of the two agents can be described via a combinatorial LP. This helps resolve positively the status of two markets left as open problems by Jain and Vazirani: the capacity allocation market in a directed graph with two source-sink pairs and the network coding market in a directed network with two sources.

Our algorithms for solving the corresponding nonlinear convex programs are fundamentally different from those obtained by Jain and Vazirani; whereas they use the primal-dual schema, our main tool is binary search powered by the strong LP-duality theorem.

1 Introduction

A convex program given by Eisenberg and Gale [EG59] captures, as its optimal solution, equilibrium allocations for the linear case of Fisher's market model. This program has several interesting properties, including the fact that it is *rational*, i.e., always has rational (as in a rational number) solutions if all the input parameters are rational. Rationality of solutions is of course an important property in mathematics; from a computer science perspective, rationality is often essential in input/output representation of values.

Recently, Jain and Vazirani [JV07] initiated a systematic study, both structural and algorithmic, of convex programs having the same basic structure as the one by Eisenberg and Gale. They called such programs, *Eisenberg-Gale-type programs*. They defined *Eisenberg-Gale markets*, or simply EG markets, as any market whose equilibrium allocations can be captured via an Eisenberg-Gale type convex program. A detailed discussion is in Section 2.

*Department of Combinatorics and Optimization, University of Waterloo, Email: deepc@math.uwaterloo.ca

†Microsoft Research, Redmond. Email: ndevanur@gmail.com

‡College of Computing, Georgia Tech. Email :vazirani@cc.gatech.edu Supported in part by NSF Grant CCF-0728640 and ONR Grant N000140910755.

Apart from the linear utilities, EG markets contain Fisher markets with more general utility functions including scalable utilities [Eis61], Leontief utilities [CV04], linear substitution utilities [Ye07] and homothetic utilities with productions [JVY05]. EG markets also contain the resource allocation market of Kelly [Kel97] used for explaining rate control and routing in multiservice networks. We describe this market in a little more detail in Section 2.2.

In their investigation of EG markets, [JV07] observed that rationality of Eisenberg-Gale-type convex programs is not limited to the original Eisenberg-Gale program – they demonstrated that several other EG-type convex programs, and hence the corresponding EG markets, are rational. On the other hand, several other EG markets are known to be irrational even when the number of agents is 3 [GJTV05, JV07]. Furthermore, [JV07] observed that most of the EG-type programs which were shown to be rational have an underlying combinatorial problem satisfying a minimax property which is crucial in proving rationality; these properties also facilitated designing strongly polynomial time algorithms to give the equilibrium prices and allocations.

[JV07] left the status of two EG markets and the corresponding EG-type programs unresolved: Kelly's market in directed networks with two agents and the *network coding market* with two agents. We describe these markets in Section 2.2. For both these markets, [JV07] showed that if the number of agents is three or more, the markets could be irrational. Moreover, they observed that for both the markets, the underlying combinatorial problem *did not* satisfy minimax theorems and were therefore expected to be irrational.

1.1 Our results and techniques

In this paper we show that despite the lack of minimax theorems for their underlying combinatorial problems, both of the markets above are rational. In fact, we prove a more general statement: *Any EG market with two agents is rational* (Theorem 2.3). As stated above, there are examples of EG markets with 3 agents which are irrational. Furthermore, if the set of feasible utilities of the two agents can be described by a *combinatorial LP* (an LP whose matrix entries have encoding size a polynomial in the dimension), then we give a strongly polynomial time algorithm to find the equilibrium prices and allocations for the EG(2) market (an EG market with 2 agents). A strongly polynomial time algorithm performs a number of elementary operations ($+, \times$,etc) that is polynomial in the number of input entries, irrespective of the values of the input entries. As a special case, we get strongly polynomial time algorithms for both the markets mentioned in the previous paragraph.
Our techniques for proving the rationality and designing the polynomial time algorithm for EG(2) markets are very different from the techniques of [JV07]. We circumvent the requirement of combinatorial minimax theorems by using the more general LP-duality theory itself; on the flip side, our methods work only for the case of two agents. Whereas [JV07] use the primal-dual schema and their algorithms can be viewed as ascending price auctions, we use a carefully constructed binary search. The algorithms of [JV07] are combinatorial whereas ours require a subroutine for solving combinatorial LP's. Tardos [Tar86] gave a strongly polynomial time algorithm to solve combinatorial linear programs.

For linear Fisher markets when restricted to 2 agents, the set of feasible utilities is not described by a combinatorial LP. Even so, we show in Section 6 that equilibrium for this market can be computed in strongly polynomial time. This raises an exciting question: is the equilibrium for all EG(2) markets, combinatorial and otherwise, computable in strongly polynomial time?

1.2 2-agent vs. multi-agent systems

Two agent systems often have special properties which do not carry over to systems with more agents. A particularly striking case of this phenomenon is Nash equilibrium: 2-player games have rational equilibria and Nash gave a beautiful example of a 3-player game that has only irrational ones [Nas50b]. From a computational viewpoint, 2-player Nash is PPAD-complete [CDT09]; however, 3-player Nash is FIXP-complete [EY07].

Clearly, the dichotomy established for Eisenberg-Gale markets in the present paper and in [JV07] is also pointing to the same phenomenon. Building on our paper, this dichotomy was extended by Vazirani [Vaz09] to Nash and nonsymmetric bargaining games [Nas50a]. [Vaz09] considers the class of these games whose solution is captured by a convex program having only linear constraints, called LNB. The restriction of LNB to the 2-player case always has a rational solution that can be found in polynomial time using only an LP solver; moreover, if all the coefficients in the program are "small" then the solution can be found in strongly polynomial time. On the other hand, there are 3-player games in LNB that have only irrational solutions, hence disallowing such algorithms.

2 Eisenberg-Gale Markets

2.1 Linear Fisher Markets and the Eisenberg-Gale Convex Program

In the linear Fisher market model, we have a set of agents I with moneys $\{m_i : i \in I\}$ and a set of divisible items J with 1 unit of item $j \in J$. Each agent has a linear utility function $u_i : \mathbf{R}^J \to \mathbf{R}$ given by $u_i(x) := \sum_{j \in J} u_{ij} x_{ij}$, where x_{ij} is the amount of item j given to agent i.

A set of prices $\{p_j : j \in J\}$ and a feasible allocation of items $\{x_{ij} : i \in I, j \in J, \ \ \forall j \in J, \ \sum_{i \in I} x_{ij} \leq 1\}$ is a *market equilibrium* if it satisfies the following three conditions:

- All positively priced items are fully sold, that is,

$$\forall j \in J, \quad p_j > 0 \Rightarrow \sum_{i \in I} x_{ij} = 1 \tag{1}$$

- Any item given to an agent must have the highest utility-to-price ratio.

$$\forall i \in I, \ j \in J, \quad x_{ij} > 0 \Rightarrow \quad u_{ij}/p_j \geq u_{ij'}/p_{j'} \tag{2}$$

- Every agent spends her initial endowment.

$$\forall i \in I, \quad \sum_{j \in J} p_j \cdot x_{ij} = m_i \tag{3}$$

The Eisenberg-Gale [EG59] convex program which captures the equilibrium allocation above is the following

$$\text{maximize} \qquad \sum_{i \in I} m_i \log f_i \ : \tag{4}$$
$$\forall i \in I, \ f_i = \sum_{j \in J} u_{ij} x_{ij}$$
$$\forall j \in J, \quad \sum_{i \in I} x_{ij} \le 1$$
$$\forall i \in I, j \in J, \quad x_{ij} \ge 0$$

In the above convex program, for every agent $i \in I$, f_i represents the total utility obtained by the agent with allocation x. The following theorem of Eisenberg and Gale [EG59] shows that the optimal solution corresponds to equilibrium allocations. The proof which we sketch below follows from Karash-Kuhn-Tucker (KKT) conditions characterizing the optimum of a convex program (see [BV06], Chapter 5, for instance, as a reference) using Lagrangean dual variables. As it turns out, these dual variables correspond to the equilibrium prices.

Theorem 2.1 *[EG59] Let (f, x) be an optimal solution to convex program (4). Then x is an equilibrium allocation for the linear Fisher market with the parameters defined above.*

Proof: The KKT conditions of optimality of convex program (4) tell us there exists dual variables $\{\alpha_i : \ i \in I\}$ and $\{p_j : \ j \in J, p_j \ge 0\}$ satisfying the following conditions:

$$\forall i \in I, j \in J, \quad -u_{ij}\alpha_i + p_j \ge 0 \ \text{ and } \ \forall i, \ \alpha_i \ge \frac{\partial}{\partial f_i}\left(\sum_{i \in I} m_i \log f_i\right) = m_i/f_i$$
$$\forall i \in I, j \in J, \ x_{ij} > 0 \Rightarrow \ -u_{ij}\alpha_i + p_j = 0 \ \text{ and } \ \forall i, f_i > 0 \ \Rightarrow \ \alpha_i = m_i/f_i$$
$$\forall j \in J, \ p_j > 0 \ \Rightarrow \ \sum_{i \in I} x_{ij} = 1$$

We now show that (p, x) satisfies the market equilibrium (ME) conditions, 1 to 3. The third KKT condition above corresponds to the ME condition 1. The first two KKT conditions imply the second ME condition 2. This is because $x_{ij} > 0 \Rightarrow u_{ij}/p_j = 1/\alpha_i \ge u_{ij'}/p_{j'}$ for any other j'. Finally, for any agent i,

$$\sum_{j \in J} p_j \cdot x_{ij} = \sum_{j \in J : x_{ij} > 0} (u_{ij}\alpha_i) x_{ij} = \alpha_i \sum_{j \in J} u_{ij} x_{ij} = \alpha_i f_i = m_i$$

□

2.2 Kelly's Capacity Allocation Market and the Network Coding Market

In Kelly's *capacity allocation market* [Kel97], we are given a directed network $G = (V, E)$ with edge capacities $c : E \to \mathbf{R}_+$. There are k agents, and the ith agent has initial endowment m_i and wants to send flow from a specified source s_i to sink t_i. The goal is to find non-negative prices (per unit flow) for edges and find flows f_i for these agents satisfying the following three equilibrium conditions:

- All flows paths are minimum priced paths from source to sink.
- Edges with positive price must be saturated, that is the total flow on it must be the capacity of the edge.
- Every agent spends their entire endowment on these flows.

Let $\mathcal{P}_i$ denote the set of paths from s_i to t_i and for all $P \in \mathcal{P}_i$, $f_i(P)$ denote the flow on path P.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} m_i \log f_i \\ \forall i: \quad f_i = & \sum_{P \in \mathcal{P}_i} f_i(P); \\ \forall e \in E: \quad & \sum_{i=1}^{n} \sum_{P \in \mathcal{P}_i} f_i(P) \le c(e) \\ \forall i, \forall P \in \mathcal{P}_i, \quad & f_i(P) \ge 0 \end{aligned}$$

It is not hard to modify the proof of Theorem 2.1 to see that the above convex program (the solution and the Lagrangean duals) gives the equilibrium flows and prices for the above market. A full proof is given in [JV07].

In the *network coding market* [JV07] we are given, as above, a directed network $G = (V, E)$ with capacities $c : E \to \mathbf{R}_+$. The set of vertices V is partitioned into two sets – terminals T and Steiner nodes S. The agents are a set of terminals, $I \subseteq T$. Every agent $i \in I$ has initial money m_i and wishes to broadcast at rate f to all terminals in T. By the network coding theorem [ACLY00] (hence the name), this is possible iff a *f-generalized branching rooted at i* is provided to the agent i. An f-generalized branching rooted at i is a subgraph of G specified by $\{b : E \to \mathbf{R}_+,\ b(e) \le c(e), \forall e \in E\}$, such that a flow of value f is possible from i to every other terminal with respect to capacities $b(e)$. A set B of generalized branchings is feasible if for each edge $\sum_{b \in B} b(e) \le c(e)$. An edge is saturated if the inequality is tight.

An equilibrium in this market is given by non-negative prices p_e for edges and for each agent i, an f_i-generalized branching b_i rooted at i such that the set $\{b_i :\ i \in I\}$ is feasible. Given p_e let the price of any generalized branching b be denoted as $p(b) := \sum_e b(e)p_e$. In an equilibrium, the following three conditions are satisfied:

- Only saturated edges have positive prices.
- For every agent i, for any f-generalized branching b rooted at i, $p(b_i)/f_i \le p(b)/f$.
- For every agent i, $p(b_i) = m_i$.

As above, a similar convex program to the above two captures the equilibrium allocations for the network coding market. Given a set $U \subseteq I$, let $\delta(U)$ denote the set of edges going from U to $V \setminus U$. Given i, let $\mathcal{U}_i$ denote the subsets $U \subseteq T$ such that $i \in U$ and $U \neq T$.

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} m_i \log f_i \\
\forall i, \forall U \in \mathcal{U}_i: \quad & f_i \leq \sum_{e \in \delta(U)} b_i(e) \\
\forall e \in E: \quad & \sum_{i \in I} b_i(e) \leq c(e) \\
& \forall i, \forall e, \quad b_i(e) \geq 0
\end{aligned}$$

Theorem 2.2 *Let (f, b) be an optimal solution to the above convex program. Then $\{b_i : i \in I\}$ is an equilibrium allocation for the network coding market.*

Proof: The KKT conditions of optimality of the above convex program tell us there exists dual variables $\{\alpha_{i,U} : i \in I, U \in \mathcal{U}_i\}$ and $\{p_e : e \in E, p_e \geq 0\}$ satisfying the following conditions:

$$\forall i \in I, e \in E, \quad -\Big(\sum_{i,U \in \mathcal{U}_i : e \in \delta(U)} \alpha_{i,U}\Big) + p_e \geq 0 \quad \text{with equality when} \quad b(e) > 0$$

$$\forall i, \quad \sum_{i,U \in \mathcal{U}_i} \alpha_{i,U} \geq \frac{\partial}{\partial f_i}\Big(\sum_{i \in I} m_i \log f_i\Big) = m_i/f_i \quad \text{with equality when} \quad f_i > 0$$

$$\forall j \in J, \ p_j > 0 \quad \Rightarrow \quad \sum_{i \in I} x_{ij} = 1$$

We now show that (p, b) satisfies the market equilibrium (ME) conditions above. The third KKT condition above corresponds to the first ME condition 1. The first two KKT conditions imply the second ME condition. This is because for any feasible f-generalized matching b rooted at i, we have

$$\begin{aligned}
p(b) = \sum_{e \in E} p_e b(e) \geq \sum_{e \in E} b(e) \cdot \Big(\sum_{i,U \in \mathcal{U}_i : e \in \delta(U)} \alpha_{i,U}\Big) & \quad \text{from KKT condition 1} \\
= \sum_{i,U \in \mathcal{U}_i} \alpha_{i,U} \Big(\sum_{e \in \delta(U)} b(e)\Big) \geq \sum_{i,U \in \mathcal{U}_i} \alpha_{i,U} f & \quad \text{from feasibility of } f
\end{aligned}$$

Moreover, in the above analysis, if b is replaced by b_i and f by f_i, the above holds with equality. Therefore, $p(b)/f \geq \sum_{i,U \in \mathcal{U}_i} \alpha_{i,U} = p(b_i)/f_i$. This is the second ME condition. The third ME condition follows from the fact that $\sum_{i,U \in \mathcal{U}_i} \alpha_{i,U} = m_i/f_i$ which along with the equality in the previous line implies $p(b_i) = m_i$. □

2.3 EG Markets

The unifying feature of all the three markets described above is that a similar convex program captures the equilibrium *utilities* of the agents. Motivated by this resemblance, Jain and Vazirani [JV07] undertook a systematic study of convex programs which they termed Eisenberg-Gale type convex programs[1].

Definition 1 *Given a matrix* $A \in \mathbf{R}^{n\times m}$ *and vectors* $b \in \mathbf{R}^m$*, the polytope* $\Pi := \{Af \leq b, \quad f \geq 0\}$ *is* down-montone in the first k coordinates *if for any feasible* $f \in \Pi$ *with first* k *coordinates* $(f_1, \cdots, f_k)$*, and for any* $(f'_1, \cdots, f'_k)$ *with* $f'_i \leq f_i$*, there exists a feasible* $f' \in \Pi$ *with the first* k *coordinates* $(f'_1, \cdots, f'_k)$*.*

Definition 2 *Given a vector* $m \in \mathbf{R}^k_+$ *and a polytope* $\Pi(A, b)$ *down-monotone in the first* k *coordinates, the following convex program is called an Eisenberg-Gale type convex program:*

$$\max\{\sum_{i=1}^{k} m_i \log f_i : \quad Af \leq b, \quad f \in \mathbf{R}^n \geq 0\}$$

Note that the objective has k *terms in its summand while* f *is in* $\mathbf{R}^n$*. We will call* k *to be the size of the EG-type convex program. We denote the program as* $CP(A, b, m, k)$*.*

Definition 3 *[JV07] An Eisenberg-Gale market, or simply EG market,* $\mathcal{M}$*, consists of a set of buyers (agents)* I *(*$|I| = k$*) such that the equilibrium* utilities $\{f_i : i \in I\}$ *of the market given moneys* $\{m_i : i \in I\}$ *is captured by an Eisenberg-Gale type convex program. That is, there exists* $A \in \mathbf{R}^{n\times m}$ *and* $b \in \mathbf{R}^m$ *such that the equilibrium utilities of the agents given money* m_i*'s is the solution to*

$$\max\{\sum_{i=1}^{k} m_i \log f_i : \quad Af \leq b, \quad f \in \mathbf{R}^n \geq 0\} \tag{5}$$

Note that there are no items *in this definition. The market will be described by* $\mathcal{M}(A, b, k)$ *and the particular instance by* $\mathcal{M}(A, b, m, k)$*. We denote the set of EG markets with* k *agents as* $EG(k)$*.*

In this paper, we show that any EG(2) market (equivalently any EG-type program of size 2) has a rational equilibrium (solution). We also give a polynomial time algorithm for finding the equilibrium (equivalently, solving the convex program). Moreover, if the entries of A (with no restriction on the size of the entries of b) have binary encoding length polynomial in the dimension of A, then the running time of this algorithm is polynomial in the dimension, that is the algorithm is strongly polynomial. We call such a market or convex program a *combinatorial* market or convex program. Note that the capacity allocation market and the network coding market are combinatorial markets. Our main theorem is the following.

Theorem 2.3 *Any EG(2) market is rational and there exists a polynomial time algorithm to find the equilibrium allocation and prices. Moreover, if the market is combinatorial, the algorithm runs in strongly polynomial time.*

[1] The definition in [JV07] is worded differently

Corollary 2.4 *Kelly's capacity allocation market and the network coding market are rational when there are only two agents and the equilibrium allocation and prices can be found in strongly polynomial time.*

In the remainder of the section we provide a roadmap to proving the above theorem. Before doing so, we state what the KKT conditions applied to the Eisenberg-Gale type convex program $CP(A, b, m, 2)$.

Suppose f is an optimal solution to the convex program (5) with $k = 2$. Then the Karash-Kuhn-Tucker conditions of optimality tells that there must be Lagrangean variables $p \in \mathbf{R}^m$ satisfying the following

- $\forall j \in [m], \quad p_j > 0 \Rightarrow \sum_{i=1}^{n} A_{ji} f_i = b_j$.
- $\forall i \in [n] \setminus \{1, 2\}, \quad f_i > 0 \Rightarrow \sum_{j=1}^{m} p_j A_{ji} = 0$ and $\sum_{j=1}^{m} p_j A_{ji} \geq 0$, otherwise.
- $i = \{1, 2\}, \quad m_i = f_i \cdot \sum_{j=1}^{m} p_j A_{ji}$.

2.4 Roadmap of the proof of Theorem 2.3

The proof of Theorem 2.3 will follow in the following three steps. Firstly, we show that instead of looking at the polytope

$$\Pi := \{f \in \mathbf{R}^n : Af \leq b, f \geq 0\}$$

we can look at the projection of the polytope onto the first two coordinates. Since the polytope Π is down-monotone in the first two coordinates and convex, in general the projection of Π on the two dimensional plane spanned by f_1 and f_2 is given by

$$\Pi_2 := \{(f_1, f_2) : f_2 \leq \beta_0; \quad f_1 + \alpha_\ell f_2 \leq \beta_\ell, 1 \leq l \leq M; \quad f_1, f_2 \geq 0\}$$

where $f_1 + \alpha_\ell f_2 = \beta_\ell$ is a facet-inducing inequality for all $1 \leq l \leq M$. We may assume WLOG that α_ℓ's are decreasing. We will call the facet $f_1 + \alpha_\ell f_2 = \beta_\ell$ the ℓth facet. We give details on the projection in Section 3.

Clearly,

$$\max\{m_1 \log f_1 + m_2 \log f_2 : f \in \Pi\} = \max\{m_1 \log f_1 + m_2 \log f_2 : (f_1, f_2) \in \Pi_2\}$$

However, applying the Karash-Kuhn-Tucker (KKT) conditions on the latter convex program gives rise to Lagrangean variables corresponding to facets of Π_2. In Section 3.3, we show how to "project up" from these prices of facets to prices p_j corresponding to rows of A. For the special case of the capacity allocation market and the network coding market, the prices of facets is only an abstract quantity while projecting up these prices leads to prices on edges.

Next we show that there exists Lagrangean duals certifying the optimal solution $\max\{m_1 \log f_1 + m_2 \log f_2 : (f_1, f_2) \in \Pi_2\}$ with the following property: either one facet ℓ is priced or two neighboring facets ℓ and $l+1$ are priced. This is determined by the ratio of the two quantities m_1 and m_2. Moreover given facets ℓ and $l+1$ and m_i's, one can check if pricing these facets leads to equilibrium. If so, the prices are just a rational function of $\alpha_\ell, \beta_\ell, m_1, m_2$ and is thus rational. We describe this in Section 4.

The above suggests the following algorithm: go over the facets one-by-one checking if one can price it (or/and its neighbor) to satisfy the KKT conditions. This will give a polynomial time algorithm if the number of facets of Π_2 is polynomial in the dimension of A. For instance, in the case of capacity allocation markets in undirected graphs, one can show using Hu's theorem [Hu63] that there are at most three facets in Π_2. However, in general the number of facets could grow exponentially. In fact even for directed networks this can be the case which is in contrast to undirected networks. Such an example is non-trivial and we give one in the appendix. To get a polynomial time algorithm we use a binary search algorithm instead where we "search" (via solving LPs) for the correct facet to price. We show that if A is combinatorial (all entries are of size a polynomial in the dimension), the binary search takes time polynomial in the dimension and by a theorem of Tardos [Tar86], the LP can be solved in time polynomial in the dimension as well. We describe this in Section 5.

3 Projection of the polytope Π

Recall the polytope specifying the feasible $f \in \mathbf{R}^n$:

$$\Pi := \{f \in \mathbf{R}^n : Af \leq b, f \geq 0\}$$

which is assumed to be down-montone in the first two dimensions. The two dimensional polytope Π_2 is obtained by projecting Π onto the plane spanned by the first two coordinates. Since Π is down-montone in the first two coordinates, positive and convex, so is Π_2. Therefore, the facets of the polytope Π_2 are line segments with increasing negative slope. In other words,

$$\Pi_2 := \{(f_1, f_2) : f_2 \leq \beta_0; \quad \forall 1 \leq l \leq M; \; f_1 + \alpha_\ell f_2 \leq \beta_\ell; \quad f_1, f_2 \geq 0\}$$

We may assume without loss of generality that α_ℓ's are decreasing. We may also assume for all $1 \leq l \leq M$, the equation $f_1 + \alpha_\ell f_2 = \beta_\ell$ is facet-inducing. We call the ℓth equality the ℓth facet. Here we point out a technicality – the equation $f_2 = \beta_0$ might or might not be facet inducing. We come to this point shortly when we describe how one obtains the facet-inducing inequalities of Π_2.

3.1 Characterization of facets

We now characterize when $f_1 + \alpha f_2 = \beta$ is a facet of Π_2, for some $\alpha, \beta \geq 0$. If it is, we say that α induces a facet. Before doing so, we make a definition. For $\alpha \geq 0$, let $L_{\Pi_2}(\alpha)$ be the linear program $\max\{f_1 + \alpha f_2 : (f_1, f_2) \in \Pi_2\}$. Also, we will let $L(\infty)$ be the linear program $\max\{f_2 : (f_1, f_2) \in \Pi_2\}$. Call $f \in \mathbf{R}^n$ an *extension* of (f_1, f_2) if its first two coordinates are (f_1, f_2) and f is in Π. Since Π_2 is just a projection of Π, $L(\alpha) = \max\{f_1 + \alpha f_2 : f \in \Pi\}$ and moreover the solution f is just an extension to (f_1, f_2). Henceforth, we will denote this LP as $L(\alpha)$, as well.

Now, since $f_1 + \alpha f_2 \leq \beta$ must be a valid inequality, if $f_1 + \alpha f_2 = \beta$ is facet inducing, we must $\beta = L(\alpha)$. Since we are in two dimensions, a valid inequality is a facet iff there are two distinct points in Π_2 which satisfy both the inequalities with equality. Thus we have the claim

Claim 3.1 *$f_1 + \alpha f_2 = \beta$ is a facet iff $\beta = L(\alpha)$ and there exists (g_1, g_2) and (h_1, h_2) in Π_2 such that $g_1 + \alpha g_2 = h_1 + \alpha h_2 = \beta$.*

The same claim also holds for the valid inequality $f_2 \leq \beta_0$ – it is a facet iff $\beta_0 = L(\infty)$ and there are two distinct points satisfying it with equality.

3.2 Finding facets

In this section we describe a procedure `FindFacet` which will be useful in our final algorithm. The procedure takes input an $\alpha > 0$ and either (A) decides that $f_1 + \alpha f_2 = L(\alpha)$ is a facet and return its two end points; or (B) returns α_ℓ and $\alpha_{\ell+1}$ such that $\alpha_{\ell+1} \leq \alpha \leq \alpha_\ell$ and α_ℓ and $\alpha_{\ell+1}$ induce neighboring facets.

Firstly observe that if α does not induce a facet, then $L(\alpha)$ has a unique solution. For otherwise we will have two distinct points satisfying $f_1 + \alpha f_2 = L(\alpha)$. Let this unique solution be g whose first two coordinates be (g_1, g_2). Now note that (g_1, g_2) must be the intersection of two neighboring facets, ℓ and $(l + 1)$. Thus, g must maximize $L(\alpha_\ell)$ and $L(\alpha_{\ell+1})$. In fact we have the following theorem.

Theorem 3.2 $\alpha_\ell = \max\{\alpha : g \textit{ maximizes } L(\alpha)\}, \quad \alpha_{\ell+1} = \min\{\alpha : g \textit{ maximizes } L(\alpha)\}$

Proof: It is enough to show that the α^* which satisfies $\max\{\alpha : g \text{ maximizes } L(\alpha)\}$ and α_* which satisfies $\min\{\alpha : g \text{ maximizes } L(\alpha)\}$ induce facets. They will be neighboring since they share a common point (g_1, g_2).

By definition of α^*, $g_1 + \alpha^* g_2 = L(\alpha^*)$. We show α^* induces a facet by exhibiting another point which satisfies $f_1 + \alpha^* f_2 = L(\alpha^*)$. Let $\delta := \min_{1 \leq l \leq M}(\alpha_\ell - \alpha_{\ell+1})$. Since A is finite, we can assume $\delta > 0$. Later in this section we give tighter bounds on how small δ can be but for the time being $\delta > 0$ suffices. Choose $0 < \epsilon < \delta$. Let (f_1, f_2) be the first two coordinates of solution to $L(\alpha^* + \epsilon)$. By definition, $(f_1, f_2) \neq (g_1, g_2)$. Also, $f_1 + (\alpha^* + \epsilon) f_2 = L(\alpha^* + \epsilon)$. Taking limits of $\epsilon \to 0$, it must be $f_1 + \alpha^* f_2 = L(\alpha)$. The proof of α_* inducing a facet is similar. $\square$

Thus, if α is not a facet, we can perform part (B) of `FindFacet` if we can solve the maximization and minimization above. We now show that α_ℓ and $\alpha_{\ell+1}$ can be found by solving two linear programs. Consider the dual of the LP $L(\alpha)$. It is the following LP, $D(\alpha)$:

$$\begin{aligned} \min\{b^T \cdot y : \quad & y^T \cdot A_1 \geq 1; \quad y^T \cdot A_2 \geq \alpha; \\ & \forall j = 3 \cdots n, \quad y^T \cdot A_j \geq 0; \quad y \geq 0\} \end{aligned}$$

where A_j is the jth column of the matrix A. Since g is an optimal solution also to $L(\alpha)$, by complementary slackness we must have that any optimal dual solution y to $D(\alpha)$ satisfies

- $y^T \cdot A_1 = 1$

- $y^T \cdot A_2 = \alpha$
- $\forall j = 3 \cdots n,\, g_j > 0 \Rightarrow y^T \cdot A_j = 0$

In fact, complementary slackness also gives us that if *any* (y, α) satisfies the above conditions with g, then g maximizes $L(\alpha)$. Therefore the following polytope, $\mathcal{Q}(g)$ precisely captures the α's for which g maximizes $L(\alpha)$. Therefore, α_ℓ and $\alpha_{\ell+1}$ can be found by maximizing and minimizing α over the polytope $\mathcal{Q}(g)$.

$$\{\alpha : \; y^T \cdot A_1 = 1, \;\; y^T \cdot A_2 = \alpha, \quad \forall j = 3 \cdots n, \;\; y^T \cdot A_j \geq 0 \\ \forall j = 3 \cdots n, g_j > 0 \Rightarrow y^T \cdot A_j = 0; \quad y \geq 0\}$$

The above discussion implies the following theorem.

Theorem 3.3 *Given $\alpha > 0$ and g maximizing $L(\alpha)$, let $\alpha_\ell = \max\{\alpha : \alpha \in \mathcal{Q}(g)\}$ and $\alpha_{l-1} = \min\{\alpha : \alpha \in \mathcal{Q}(g)\}$. Then $\alpha_{l-1} \leq \alpha \leq \alpha_\ell$ and α_{l-1} and α_ℓ induce neighboring facets.*

If α satisfies any of the inequalities in Theorem 3.3 with equality then it induces a facet. Otherwise it doesn't. We state a theorem about the *granularity* of the α_ℓ's which induce facets of Π_2. Given any rational number α, let $\nu(\alpha)$ denote the number of bits in a binary encoding of α. Such a binary encoding could be the binary representation of the numerator and the denominator. Given a matrix A, let $\nu(A)$ be the number of bits required to encode it. That is, $\nu(A)$ is $\sum_{i,j} \nu(A_{ij})$ where A_{ij} is the ijth entry of the matrix, assumed to be rational. Recall a matrix A is combinatorial if $\nu(A)$ is a polynomial in the dimensions of A.

Theorem 3.4 *There exists a constant c depending only on the dimension of A such that for every facet α_ℓ of Π_2, we have $\nu(\alpha_\ell) = O(\nu(A)^c)$.*

Proof: The above discussion implies that the α_ℓ's that induce facets are solutions to a linear program for which the entry matrix is A (refer description of $\mathcal{Q}(g)$ above). It follows from LP theory that the size of the binary encoding of a solution to an LP is bounded by a polynomial of the dimension of the matrix giving $\nu(\alpha_\ell) = O(\nu(A)^c)$ for some constant c depending only on dimension of A. Moreover, if $\nu(A)$ is a polynomial in the dimension of A, so is $\nu(\alpha_\ell)$ for all ℓ. □

Theorem 3.5 *There exists constants K and ϵ depending only on A, such that $\alpha_1 \leq K$ and for any ℓ, $\alpha_\ell - \alpha_{\ell+1} \geq \epsilon$.*

Proof: Let K be the largest integer with encoding $O(\nu(A)^c)$ where c is as in the above theorem. In particular, K is larger than the numerators and denominators of the rational numbers α_ℓ. In particular, $K \geq \alpha_1$. Moreover, if we choose $\epsilon = 1/K^2$, since α_ℓ and $\alpha_{\ell+1}$ are rational numbers with denominators at most K, their difference is larger than ϵ. □

Now we describe given an α which induces a facet, how to find its end points. This will complete the procedure we started in this section.

Theorem 3.6 *Let $\alpha \geq 0$ induce a facet of Π_2. Let (g_1, g_2) be the solution to $L(\alpha + \epsilon)$ and (h_1, h_2) be the solution to $L(\alpha - \epsilon)$. Then (g_1, g_2) and (h_1, h_2) are the end points of the facet $f_1 + \alpha f_2 = L(\alpha)$.*

Proof: Suppose $\alpha = \alpha_\ell$ is the ℓth facet. From Theorem 3.5 it follows that neither $(\alpha + \epsilon)$ nor $(\alpha - \epsilon)$ induce facets. Moreover, $\alpha_{l-1} > (\alpha + \epsilon) > \alpha_\ell > (\alpha - \epsilon) > \alpha_{\ell+1}$. Therefore, $L(\alpha + \epsilon)$ is maximized uniquely by the intersection of the facets induced by α_{l-1} and α_ℓ, while $L(\alpha - \epsilon)$ is maximized uniquely by the intersection of the facets induced by α_ℓ and $\alpha_{\ell+1}$. □

The following theorem summarizes the procedure `FindFacet` which will be used in Section 5.

Theorem 3.7 *The procedure* `FindFacet` *takes input a rational $\alpha > 0$ and either*

> *(A) decides that $f_1 + \alpha f_2 = L(\alpha)$ is a facet and return its two end points; or*
> *(B) returns α_ℓ and $\alpha_{\ell+1}$ such that $\alpha_{\ell+1} \leq \alpha \leq \alpha_\ell$ and α_ℓ and $\alpha_{\ell+1}$ induce neighboring facets, and returns the endpoints of these facets.*

The time taken by the procedure is $T(A)$ which a polynomial in $\nu(A) + \nu(b)$ and just a polynomial in $\nu(A)$ if $\nu(A)$ is a polynomial in the dimension of A.

Proof: The proof of correctness of the procedure follows from the discussion preceding it. The fact that $T(A)$ is a polynomial in $\nu(A)$ follows from an algorithm of Tardos [Tar86] to solve a linear program with entries of size polynomial in the dimension, in time a polynomial in $\nu(A)$. □

3.3 Pricing of Facets

Let us now review the KKT conditions for the convex program:

$$\max\{m_1 \log f_1 + m_2 \log f_2 : (f_1, f_2) \in \Pi_2\} \tag{6}$$

If (f_1^*, f_2^*) is the optimal solution, then there must exist facet prices $q_0, q_1, \cdots, q_M$ such that:

$$\forall l = 0, 1, \cdots, M, \quad q_\ell > 0 \Rightarrow f_1^* + \alpha_\ell f_2^* = \beta_\ell \tag{7}$$

$$m_1 = f_1^* \cdot \sum_{l=1}^{M} q_\ell; \qquad m_2 = f_2^* \cdot (q_0 + \sum_{l=1}^{M} \alpha_\ell q_\ell)$$

In Section 4, we will use the above inequalities to obtain the prices of the facets in terms of m_1 and m_2. In the remainder of the section we show how to "project up" the solution (f_1^*, f_2^*) and $(q_1, \ldots, q_M)$ to a solution for Π and their corresponding Lagrangean dual variables. That is, we show how to obtain an extension f of (f_1^*, f_2^*) and the Lagrangean prices for the original program, $(p_1, \cdots, p_m)$, which satisfy the KKT conditions of the original program, namely:

$$\forall j \in [m], \quad p_j > 0 \Rightarrow \sum_{i=1}^{n} A_{ji} f_i = b_j \tag{8}$$

$$\forall i \in [n] \setminus \{1,2\}, \quad f_i > 0 \Rightarrow \sum_{j=1}^{m} p_j A_{ji} = 0 \text{ and } \sum_{j=1}^{m} p_j A_{ji} \geq 0, \text{ otherwise}$$

$$\text{for } i = \{1,2\}, \quad m_i = f_i \cdot \sum_{j=1}^{m} p_j A_{ji}$$

For every ℓ such that $q_\ell > 0$, consider the LP:

$$\max\{f_1 + \alpha_\ell f_2; \quad Af \leq b; \quad f \geq 0\}$$

From equation (7) it follows that the optimum of the above LP must be $\beta_\ell = f_1^* + \alpha_\ell f_2^*$. Therefore, any extension f of (f_1^*, f_2^*) is an optimum solution to the above LP. Consider the dual of the above LP

$$\min\{b^T \cdot y : \quad y^T \cdot A_1 \geq 1; \quad y^T \cdot A_2 \geq \alpha; \quad \forall i = 3, \cdots, n; \; y^T \cdot A_i \geq 0; \quad y \geq 0\} \tag{9}$$

and for every ℓ consider any optimal solution $y^{(\ell)}$ to the above dual. Define $p_j = \sum_{l=1}^{M} y_j^{(\ell)} q_\ell$ for $j = 1 \cdots m$.

Theorem 3.8 $(f, p_1, \cdots, p_m)$ *satisfy the original KKT conditions* (8). *Moreover they can be calculated in time a polynomial in the encoding size* $\nu(A)$ *of the matrix* A.

Proof: Note that $p_j > 0$ implies $y_j^{(\ell)} > 0$ for some ℓ. By complementary slackness this implies $\sum_{i=1}^{n} A_{ji} f_i = b_j$. Similarly, by complementary slackness, for $i \in [n] \setminus \{1,2\}$, $f_i > 0 \Rightarrow \sum_{j=1}^{m} y_j^{(\ell)} A_{ji} = 0$, for every ℓ such that $q_\ell > 0$. Thus,

$$\sum_{j=1}^{m} p_j A_{ji} = \sum_{j=1}^{m} (\sum_{l=1}^{M} y_j^{(\ell)} q_\ell) A_{ji} = \sum_{l=1}^{M} q_\ell (\sum_{j=1}^{m} y_j^{(\ell)} A_{ji}) = 0$$

Lastly, $f_1^* > 0$ implies $\sum_{j=1}^{m} y_j^{(\ell)} A_{j1} = 1$ for all $l : q_\ell > 0$. Moreover if $q_\ell > 0$ we have $m_1 = f_1^* \sum_{l=1}^{M} q_\ell$. The two equations imply $m_1 = \sum_{j=1}^{m} p_j A_{j1}$. The result for m_2 holds similarly.

The time taken to find $p_1, \cdots, p_m$ is the time taken to solve (9) which is a polynomial in $\nu(A)$ (Note that the right hand side of this LP is only 0 or 1). $\square$

4 Rationality of EG(2) markets

In this section we show that for any matrix A with rational entries and rational numbers m_1, m_2 the solution f to (5) and the Lagrangean prices $p_1, \cdots, p_m$ are rational. From Theorem 3.8 it suffices to show that the optimum and Lagrangean duals $q_1, \cdots, q_M$ to (6), are rational.

Let us recall the KKT conditions (7) for the optimality of (f_1, f_2) for (6)

$$\forall l = 0, 1, \cdots, M, \quad q_\ell > 0 \Rightarrow f_1 + \alpha_\ell f_2 = \beta_\ell$$
$$m_1 = f_1 \cdot \sum_{l=1}^{M} q_\ell; \qquad m_2 = f_2 \cdot (q_0 + \sum_{l=1}^{M} \alpha_\ell q_\ell)$$

Claim 4.1 *For any m_1, m_2, at most two of the q_ℓ's are positive.*

Proof: From the first condition any $q_\ell > 0$ implies an equation in two variables. Since all of these are linearly independent (they induce distinct facets), at most two equations can be satisfied by (f_1, f_2). □

Given the ℓth facet induced by α_ℓ, let (g_1^ℓ, g_2^ℓ) and (h_1^ℓ, h_2^ℓ) denote the two end points of the facet. Without loss of generality we will assume $g_1^\ell \leq h_1^\ell$ and $g_2^\ell \geq h_2^\ell$. Note that $(h_1^\ell, h_2^\ell) = (g_1^{\ell+1}, g_2^{\ell+1})$. Now we divide the interval $[0, 1]$ into the following $2M - 2$ intervals:

Definition 4 *For $1 \leq l < M$:*

$$I_\ell := \left[\frac{g_1^\ell}{L(\alpha_\ell)}, \frac{h_1^\ell}{L(\alpha_\ell)} \right]; \quad I_{\ell,\ell+1} := \left[\frac{h_1^\ell}{L(\alpha_\ell)}, \frac{h_1^\ell}{L(\alpha_{\ell+1})} \right]$$

If $g_1^1 > 0$ (implying $f_2 = \beta_0$ is a facet), then

$$I_{0,1} = \left[0, \frac{g_1^\ell}{L(\alpha_1)} \right]$$

Claim 4.2 *Each of the intervals defined above are disjoint and cover* $[0, 1]$.

Proof: Note that for each I_ℓ, $I_{\ell,\ell+1}$, the right end point is larger than the left and moreover the right end point of I_ℓ is the same as the left of $I_{\ell,\ell+1}$. Moreover the left end point of $I_{0,1}$ or I_1 is 0. Also, $h_1^M = \beta_M = L(\alpha_M)$ implying the right end point of $I_{M-1,M}$ is 1. □

The following theorem gives us the one or two facets which we should price.

Theorem 4.3 *Let $\rho = \frac{m_1}{m_1 + m_2}$.*

1. *If $\rho \in I_\ell$ for $l > 0$, $q_\ell = \frac{m_1 + m_2}{L(\alpha_\ell)}$ and $f_1 = m_1/q_\ell$ and $f_2 = m_2/(\alpha_\ell q_\ell)$ satisfy KKT conditions* (7)*.*

2. *If $\rho \in I_{\ell,\ell+1}$ for $l > 0$, then $(f_1, f_2) = (h_1^\ell, h_2^\ell)$ and*

$$q_\ell = \frac{h_1^\ell m_2 - \alpha_{\ell+1} h_2^\ell m_1}{h_1^\ell h_2^\ell (\alpha_\ell - \alpha_{\ell+1})}; \quad q_{l+1} = \frac{\alpha_\ell h_2^\ell m_1 - h_1^\ell m_2}{h_1^\ell h_2^\ell (\alpha_\ell - \alpha_{\ell+1})}$$

   satisfy KKT conditions (7)*.*

3. *If $\rho \in I_{0,1}$ (that is $f_2 = L(\infty)$ is a facet), then $(f_1, f_2) = (h_1^0, L(\infty))$ and*

$$q_0 = \frac{h_1^0 m_2 - \alpha_1 L(\infty) m_1}{h_1^0 L(\infty)}; \quad q_1 = \frac{m_1}{h_1^0}$$

   satisfy KKT conditions (7)*.*

Proof:

1. Observe that $q_\ell > 0$ and that $f_1 + \alpha_\ell f_2 = \frac{m_1+m_2}{q_\ell} = L(\alpha_\ell)$. Also the second KKT condition is immediate for both m_1 and m_2.

2. Check that since $\rho \in I_{\ell,\ell+1}$, both q_ℓ and q_{l+1} are non-negative. This is because

$$m_1/(m_1+m_2) \geq h_1^\ell/L(\alpha_\ell) = h_1^\ell/(h_1^\ell + \alpha_\ell h_2^\ell), \textit{and}$$
$$m_1/(m_1+m_2) \leq h_1^\ell/L(\alpha_{\ell+1}) = h_1^\ell/(h_1^\ell + \alpha_{\ell+1} h_2^\ell)$$

   which imply $h_1^\ell/(\alpha_\ell h_2^\ell) \leq m_1/m_2 \leq h_1^\ell/(\alpha_{\ell+1} h_2^\ell)$ and thus $q_\ell, q_{l+1} \geq 0$.

   It is also a calculation to check $m_1 = h_1^\ell(q_\ell + q_{l+1})$ and $m_2 = h_2^\ell(\alpha_\ell q_\ell + \alpha_{\ell+1} q_{l+1})$ and we omit it.

3. It is easy to check the KKT conditions for this case. What requires a little work is to see that $q_0 \geq 0$. Since $\rho \in I_{0,1}$, we get

$$m_1/(m_1+m_2) \leq g_1^1/L(\alpha_1) = h_1^0/L(\alpha_1) \text{ implying}$$
$$m_1/m_2 \leq h_1^0/(L(\alpha_1) - h_1^0) = h_1^0/(\alpha_1 L(\infty))$$

   where the last equality follows from the fact that $(h_1^0, L(\infty))$ lies on the facet induced by α_1 (it is an end point of it).This implies that $q_0 \geq 0$.

□

Corollary 4.4 *The equilibrium solution of any EG(2) market is rational.*

Proof: Theorem 4.3 immediately implies the equilibrium utilities (f_1, f_2) are rational. It also gives that the facet prices are rational. By Theorem 3.8, the equilibrium prices of the EG(2) market are just rational combinations of the facet prices and hence rational. □

The above also suggests an algorithm which runs in time polynomial in the number of facets. However in Appendix A we show that even for the directed two source-sink flow market, the number of facets could be exponentially larger than the number of edges. In the next section we show a binary search algorithm which is efficient.

5 Algorithms for EG(2) markets

We now give an algorithm which finds the prices q_ℓ and flow (f_1, f_2) satisfying KKT conditions (7). Using Theorem 3.8 and the fact that at most two facets are priced (which gives us an efficient way of getting p_j's from q_ℓ), we will finish the proof of our main theorem, Theorem 2.3.

We use the machinery developed in Section 3.2 to develop our binary search algorithm. To remind, we developed a procedure, `FindFacet`, which given α either decides if α induces a facet and returns its end points, or returns α_ℓ and $\alpha_{\ell+1}$ which are neighboring facets and $\alpha_\ell > \alpha > \alpha_{\ell+1}$.

From Theorem 4.3, we will be done if we find the interval I_ℓ or $I_{\ell,\ell+1}$ in which $\rho = \frac{m_1}{m_1+m_2}$ lies. However note that I_ℓ requires the knowledge of α_ℓ which we do not have as input. From Theorem 3.5, we know that α_ℓ's lie between K and 0. The algorithm starts with a guess $\alpha = K/2$ of α_ℓ. It runs `FindFacet` in which case it either gets the end points (and thus gets I_ℓ) or gets α_ℓ and $\alpha_{\ell+1}$ (and the end points of the facets induced by them) to get $I_\ell, I_{\ell,\ell+1}, I_{l+1}$. If ρ lies in these intervals, then we are done. If ρ lies to the left of I_ℓ, then we search for α in the first half; and otherwise we search for α in the second half. The procedure takes time polynomial in $\log(K)$ which is a polynomial in the description size of A. Details are in Algorithm 1 below.

Algorithm 1 Binary Search Algorithm for finding facet prices

Input: m_1, m_2, K, ϵ (We assume K, ϵ have been evaluated as described in Theorem 3.5).
Let $\rho = m_1/(m_1 + m_2)$.
Initialize $U = K$, $L = 0$.

Repeat until $U - L \leq \epsilon$

1. $\alpha = (U + L)/2$.
2. Use `FindFacet` (described in Section 3.2) to check if α induces a facet or find $\alpha_\ell, \alpha_{\ell+1}$ which do. Find the end points of these facets as well.
3. Use the end points and the α_ℓ's to get the intervals $I_\ell, I_{\ell,\ell+1}, I_{l+1}$ as described in Section 4.
4. If $\rho \in I_\ell \cup I_{\ell,\ell+1} \cup I_{l+1}$, find prices q_ℓ, q_{l+1} as described in Theorem 4.3. Use Theorem 3.8 to get the solution and prices for the original problem and exit.
5. If $\rho < I_\ell$ (that is less than the left end point of I_ℓ), $L = \alpha_\ell$.
   Else $U = \alpha_{\ell+1}$.

Let $T(A)$ be the time taken by the procedure `FindFacet`. By Theorem 3.7 it is bounded by a polynomial in $\nu(A)$ and moreover if the entries of A is bounded by a polynomial in the dimension of the matrix, then $T(A)$ is a polynomial in the dimension of the matrix as well. The following theorem completes the proof of Theorem 2.3.

Theorem 5.1 *Algorithm 1 always outputs the equilibrium prices of facets and runs in time $O\left(T(A)\log\left(\frac{K}{\epsilon}\right)\right)$ and is thus a polynomial time algorithm. If $\nu(A)$ is bounded by a polynomial in the dimension of the matrix, then the Algorithm 1 is a strongly polynomial time algorithm.*

Proof: Suppose $\frac{m_1}{m_1+m_2} \in I_k \cup I_{k-1,k}$. Then it is clear that $L \leq \alpha_k \leq U$ throughout the algorithm.

Suppose that at the end of an iteration, $U - L < \epsilon$. Note that after each iteration, either both U and L have a value equal to one of the α_ℓ's, or one of them is 0 or K and the other has a value equal to an α_ℓ. In either case, $U - L < \epsilon \implies U = L$ (follows from the definition of ϵ in Theorem 3.5), which should equal α_k by the first part. Hence we must have found the equilibrium prices in this iteration.

The number of iterations of the repeat loop is bounded by $O\left(\log\left(\frac{K}{\epsilon}\right)\right)$ since we halve the gap between U and L till the gap is smaller than ϵ. Since `FindFacet` can be done in $T(A)$ time, the theorem follows. □

Theorem 5.2 *(Restatement of Main Theorem 2.3) Any EG(2) market is rational and there exists a polynomial time algorithm to find the equilibrium allocation and prices. Moreover, if the EG(2) market is combinatorial, the algorithm runs in strongly polynomial time.*

Proof: The proof of rationality follows from Corollary 4.4. The algorithm to find prices for the EG(2) market is to run Algorithm 1 to get equilibrium utilities and facet prices and then use Theorem 3.8 to get the original prices from the facet prices.

By Theorem 5.1 and Theorem 3.8, both steps run in polynomial time. Furthermore, if the market is combinatorial both steps run in strongly polynomial time. □

6 Strongly polynomial time algorithm for linear Fisher market with two agents

Note that the set of feasible utilities of two agents in a linear Fisher market is not described via a combinatorial LP – the corresponding matrix A in convex program (5) can have entries much larger than the dimension. Nevertheless, we can obtain a strongly polynomial time algorithm for this case.

Recall, the agents have moneys m_1 and m_2 and have utility u_{1j} and u_{2j} for $j \in J$. Suppose $|J| = m$. Also recall that the equilibrium prices $\{p_j : j \in J\}$ and the allocations $\{x_{1j}, x_{2j} : j \in J\}$ must satisfy:

- For $i = 1, 2$, $x_{ij} > 0 \Rightarrow u_{ij}/p_j \geq u_{ij'}/p_{j'}, \quad \forall j' \in J$. The ratio u_{ij}/p_j is called the bang-per-buck of item j for agent i.
- $p_j > 0 \quad \Rightarrow \quad x_{1j} + x_{2j} = 1$.
- For $i = 1, 2$, $\sum_{j \in J} p_j \cdot x_{ij} = m_i$.

Sort the items in decreasing order of $\rho_j := u_{1j}/u_{2j}$. Without loss of generality, assume $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_m$. In an equilibrium solution, call an item j *shared* if for both $i = 1, 2$, $u_{ij}/p_j \geq u_{ij'}/p_{j'}$ for all items j'. That is, for both these agents, j has the highest bang-per-buck among all items. Thus, both items can get non-zero amounts of j in the equilibrium.

Claim 6.1 *In an equilibrium solution, all shared items j have the same, $\rho_j = \rho$. Moreover, for every item $\rho_\ell < \rho$, $x_{1l} = 1$ and for every item $\rho_\ell > \rho$, $x_{2l} = 1$.*

Proof: Let (x, p) be an equilibrium solution. Suppose two items j and j' are shared. Then by definition, $u_{ij}/p_j = u_{ij'}/p_{j'}$ for both $i = 1, 2$. Thus, $u_{1j}/u_{1j'} = u_{2j}/u_{2j'} = p_j/p_{j'}$. This implies, $\rho_j = \rho_{j'}$. Thus all shared items have the same ρ.

Consider an item ℓ with $\rho_\ell > \rho_j$. Suppose $x_{2l} > 0$. Then we must have $u_{2l}/p_\ell = u_{2j}/p_j$ since both items can go to agent 2. Multiplying by ρ_ℓ on the left and ρ_j on the right we get, $u_{1l}/p_\ell > u_{1j}/p_j$. But this is not possible since j is shared and therefore $x_{1j} > 0$. Thus, $x_{2l} = 0$ implying $x_{1l} = 1$. The other case is similar. □

Claim 6.2 *If there are two items j and j' with $\rho_j = \rho_{j'}$ such that x_{1j} and $x_{2j'}$ are both strictly greater than 0, then j and j' are shared.*

Proof: Since $x_{1j} > 0$, j has the highest bang-per-buck for agent 1. Thus, $u_{1j}/p_j \geq u_{1j'}/p_{j'}$. Multiplying by $1/\rho_j$ on the left and $1/\rho_{j'}$ on the right gives $u_{2j}/p_j \geq u_{2j'}/p_{j'}$. But, agent 2 has the highest bang-per-buck for item j' implying the last inequality is an equality. Therefore agent 2 has the highest bang-per-buck for item j as well, implying j is shared. The proof of j' being shared is similar. □

Given ρ, consider the set of items $A(\rho) := \{j : \rho_j > \rho\}$, $X(\rho) = \{j : \rho_j = \rho\}$, and $B(\rho) := \{j : \rho_j < \rho\}$. At an equilibrium two cases can occur.
Case 1: There are no shared items. By Claim 6.2 this implies items having the same ρ have to go completely to one agent. This implies there exists a ρ such that agent 1 gets all the items with $\rho_j \geq \rho$ $(A(\rho) \cup X(\rho))$ and agent 2 gets all items with $\rho_j < \rho$ $(B(\rho))$.
Case 2: There are shared items. By Claim 6.1, this implies there exists a ρ such that $A(\rho)$ goes to agent 1, $B(\rho)$ goes to agent 2 and items in $X(\rho)$ can go to either agent in any fraction. We can use this to get the following algorithm.

Algorithm 2 Strongly polynomial time algorithm for linear Fisher markets with 2 agents.

For items $j = 1 \ldots m$, let $\rho = \rho_j$. Construct the sets A, X, B as stated above (we remove the dependency on ρ).

(a) **Case 1**: Let p_1 be the price of item 1 and p_m be the price of item m. This fixes the prices of all items in terms of p_1, p_m – for $j \in A \cup X$, $p_j = \frac{u_{1j}}{u_{11}} p_1$ and $j \in B$, $p_j = \frac{u_{2j}}{u_{2m}} p_m$. Using the above, solve for p_1 and p_m with the equations $\sum_{j \in A \cup X} p_j = m_1$ and $\sum_{j \in B} p_j = m_2$. If $\frac{u_{11}}{p_1} \geq \frac{u_{1m}}{p_m}$ and $\frac{u_{2m}}{p_m} \geq \frac{u_{21}}{p_1}$, return this as the equilibrium solution. Else, go to the next case.

(b) **Case 2**: Pick any item ℓ in X arbitrarily. Let its price be p_ℓ. This fixes the price of all items since ℓ is shared – For items in $j \in A$, $p_j = \frac{u_{1j}}{u_{1l}} p_\ell$ and for items $j \in B$, $p_j = \frac{u_{2j}}{u_{2l}} p_\ell$. For items in $j \in X$, we have $p_j = \frac{u_{1j}}{u_{1l}} p_\ell = \frac{u_{2j}}{u_{2l}} p_\ell$, where the second equality follows from $\rho_j = \rho_\ell$. Using this, solve for p_ℓ using $\sum_{j \in A \cup X \cup B} p_j = m_1 + m_2$. Check if $\sum_{j \in A} p_j \leq m_1$ and $\sum_{j \in B} p_j \leq m_2$. If both are true, then since the items in J are shared we can find fractions in which they are divided so that the equilibrium condition is satisfied. If either of them is untrue, go to the next iteration.

Theorem 6.3 *Algorithm 2 is a strongly polynomial time algorithm for linear Fisher markets with two agents.*

Proof: Since we go over all possible values of ρ and check for the two possible cases in an equilibrium, one of these must satisfy the equilibrium condition. The time taken by the above algorithm is $O(m^2)$ – for every item, we run the two cases which takes time m each – and is thus strongly polynomial. □

7 Conclusions

In this paper, we studied Eisenberg-Gale markets with two agents and showed they always have rational equilibrium. EG markets with three agents are known to have irrational equilibrium. We also show if the feasible allocations in the market can be described via a combinatorial polytope, there is a strongly polynomial time algorithm to find the equilibrium allocation and prices. As an example, we get strongly polynomial time algorithm for Kelly's resource allocation markets with two source-sink pairs, and the network coding market in directed networks, with two sources. We also show a strongly polynomial time algorithm for the linear Fisher market with two agents; such markets are not combinatorial. Can this result be extended to all EG(2) markets?

References

[ACLY00] R. Ahlswede, N. Cai, S.Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. on Information Theory*, 46:1204–1216, 2000.

[BV06] S. Boyd and L. Vanderberghe. *Convex Optimization*. Cambridge University Press, 2006.

[CDT09] X. Chen, X. Deng, and S-H. Teng. Setlling the Complexity of Computing Two-Player Nash Equilibria. *Journal of the ACM*, 56(3), 2009.

[CV04] B. Codenotti and K. Varadarajan. Efficient computation of equilibrium prices for markets with Leontief utilities. In *Proceedings of International Colloquium on Algorithms, Logic and Programming*, 371–382, 2004.

[EG59] E. Eisenberg and D. Gale. Consensus of subjective probabilities: the Pari-Mutuel method. *The Annals of Mathematical Statistics*, 30:165–168, 1959.

[Eis61] E. Eisenberg. Aggregation of utility functions. *Management Sciences*, 7:337–350, 1961.

[EY07] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM J. on Computing*, to appear. Preliminary version in *FOCS*, 2007.

[GJTV05] D. Garg, K. Jain, K. Talwar, and V. Vazirani. A primal-dual algorithm for computing Fisher equilibrium in absence of gross substitutability property. In *Theoretical Computer Science*, 378(2):143–152, 2005.

[Hu63] T.C. Hu. Multicommodity network flows. *Operations Research*, 14:344–360, 1963.

[JV07] K. Jain and V. V. Vazirani. Eisenberg-gale markets: Algorithms and structural properties. In *Proceedings of ACM Symposium on Theory of Computing*, 364–373, 2007. To appear in *Games and Economic Behavior.*

[JVY05] K. Jain, V. V. Vazirani, and Y. Ye. Market equilibrium for homothetic, quasi-concave utilities and economies of scale in production. In *Proceedings of ACM Symposium on Discrete Algorithms*, 63–71, 2005.

[Kel97] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[Nas50a] J. F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[Nas50b] J. F. Nash. Equilibrium points in n-person games. *Porceedings National Academy of Sciences*, 36:48–49, 1950.

[Tar86] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986.

[Vaz09] V. V. Vazirani. 2-player Nash and nonsymmetric bargaining games: algorithms and structural properties. Submitted, 2009.

[Ye07] Y. Ye. Exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. *Theoretical Computer Science*, 378(2):134–142, 2007.

A Projection of two source two sink flow in directed networks

Consider a network $N(V, A)$ with two source-sink pairs (s_1, t_1) and (s_2, t_2) and capacities $c : A \rightarrow \mathbf{R}_+$. Let n be the number of vertices and m be the number of arcs. We are interested in the flow polytope which has variable f_1, f_2 indicating the total value of flow from s_1 to t_1 and s_2 to t_2 respectively. Also, there is a flow variable on each arc indicating the flow on the arc with the total flow upper bounded by the capacity c of the arc. Finally there is a conservation of flow at each vertex – the total flow coming into a vertex which is not a source or a sink equals the total flow going out. Call this polytope Π. Let Π_2 be the projection of this polytope on to the plane spanned by f_1 and f_2. The question we ask is how many facets does Π_2 have? In this section we show that Π_2 can have an exponential (in n and m) number of facets in a directed network. This result is in contrast to the case of undirected graphs where the similar Π_2 has at most 3 facets. This result follows from Hu's theorem [Hu63] on two source-sink pair flows in undirected graphs.

A description of Π_2 is the following. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ denote the set of paths from s_1 to

t_1 and s_2 to t_2 respectively. Then

$$\begin{aligned} \Pi_2(N) := \qquad & \{(f_1, f_2) : f_i = \sum_{P \in \mathcal{P}_i} f_i(P) \text{ for } i = 1, 2; \\ & \forall e \in A : \sum_{P \in \mathcal{P}_1 : e \in P} f_1(P) + \sum_{Q \in \mathcal{P}_2 : e \in Q} f_1(Q) \le c(e) \\ & f_1(P), f_2(Q) \ge 0, \quad \forall P \in \mathcal{P}_1, Q \in \mathcal{P}_2\} \end{aligned} \tag{10}$$

However all the inequalities above might not induce facets. Indeed since Π_2 is two dimensional we can describe all the facet inducing inequalities as follows:

$$\Pi_2(N) := \{(f_1, f_2) : \forall 1 \le l \le M : \quad f_1 + \alpha_\ell f_2 \le \beta_\ell; \quad f_1, f_2 \ge 0\}$$

where we may assume $\infty \ge \alpha_1 \ge \cdots \alpha_M \ge 0$, where by $\alpha_1 = \infty$ we mean the inequality $f_2 \le \beta$ is a facet. Given a network N, we call the numbers $(\alpha_1, \cdots, \alpha_M)$ the *profile* of N.

Given a network N, we denote the LP, $\max\{f_1 + \alpha f_2 : (f_1, f_2) \in \Pi_2(N)\}$, and its value by $L_N(\alpha)$. We assume $\Pi_2(N)$ is as in Equation 10. By duality we know that $L_N(\alpha)$ equals the following dual program $D_N(\alpha)$.

$$\min\{\sum_{e \in A} c(e)x(e) : \quad \forall P \in \mathcal{P}_1;\ x(P) \ge 1; \quad \forall Q \in \mathcal{P}_2;\ x(Q) \ge \alpha; \quad \forall e \in E;\ x(e) \ge 0\}$$

We recall the following characterization of a facet.

Theorem A.1 *$f_1 + \alpha f_2 = \beta$ is a facet iff $\beta = L_N(\alpha)$ and there exists two distinct feasible flows (g_1, g_2) and (h_1, h_2) satisfying the inequality with equality.*

To demonstrate our example we construct two operations. The first called the *doubling operation* takes a network N with profile $(\alpha_1, \cdots, \alpha_k)$ with $\alpha_k \ge 1$ and returns a network N' with a constant number of more arcs and vertices whose profile is $(\alpha_1, \cdots, \alpha_k, \zeta_k, \cdots, \zeta_1)$ where $\zeta_i := \frac{\alpha_i}{2\alpha_i - 1}$. Thus, N' has double the number of facets as N but only a constant number of edges more. The next operation called the *shifting operation* takes a network N with profile $(\alpha_1, \cdots, \alpha_k)$ with $\alpha_k \ge 1$ and returns a network N' with a constant number of more arcs and vertices whose profile is $(\alpha_1 + 1, \cdots, \alpha_k + 1)$. Therefore, starting with any network N with a constant number of edges, applying m steps of doubling and shifting alternately gives us a network N with $O(m)$ edges but at least 2^m facets which completes the example. In the remainder of the section we describe the two operations.

Doubling Operation:
Given a network N, Figure 1 shows how the network N' is constructed.

Lemma A.2 *Suppose the profile of N was $(\alpha_1, \cdots, \alpha_k)$ with $\alpha_k \ge 1$. Then the profile of N' is $(\alpha_1, \cdots, \alpha_k, \zeta_k, \cdots, \zeta_1, 0)$ where $\zeta_i = \frac{\alpha_i}{2\alpha_i - 1}$.*

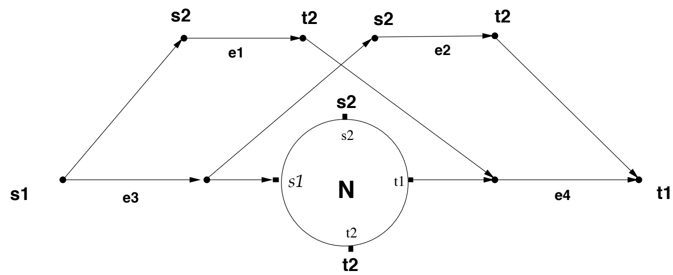

Figure 1: The network N' obtained from N. The edges e_i have a capacity C, where C is the maximum f_1 flow that is feasible in N.

Proof:

We prove the lemma by giving for each α_i (and ζ_j) two feasible flows (g_1, g_2) and (h_1, h_2) on the facet and a cut of value $L_{N'}(\alpha_i)$ (and $L_{N'}(\zeta_j)$).

Since $(\alpha_1, \cdots, \alpha_k)$ is the profile of N, for every α_i there are feasible flows of value (g_1, g_2) and (h_1, h_2) such that both satisfy $f_1 + \alpha_i f_2 = L_N(\alpha_i)$. Moreover there exists a solution $x(e)$ to $D_N(\alpha_i)$ of value $L_N(\alpha_i)$. We now describe the feasible flows of value (g_1', g_2') and (h_1', h_2') in N' satisfying $f_1' + \alpha_i f_2' = L_{N'}(\alpha_i)$ and also dual solutions $y(e)$ to $D_{N'}(\alpha_i)$ of value $L_{N'}(\alpha_i)$.

Let C be the maximum f_1 flow that can be sent in N. Note that $C = L_N(\alpha_k)$

$$g_1' = g_1, \; g_2' = g_2 + 2C; \qquad h_1' = h_1, \; h_2' = h_2 + 2C$$
$$\forall e \in E[N]; \;\; y(e) = x(e); \;\; y(e_1) = y(e_2) = \alpha_i; \;\; y(e_3) = y(e_4) = 0$$

Claim A.3 *There exist feasible flows in N' if value (g_1', g_2') and (h_1', h_2').*

Proof: We know there exists a flow of value (g_1, g_2) from s_1 to t_1 and s_2 to t_2 in N. The same flow passing via e_3 and e_4 gives a flow of g_1' in N' from s_1 to t_1 in N'. The extra $2C$ flow from s_2 to t_2 in N' is via the arcs e_1 and e_2. □

Claim A.4 *The $y(e)$'s forms a feasible solution to the dual program $D_{N'}(\alpha_i)$.*

Proof: Every s_1, t_1 path P passes through N or uses the edge e_1 or e_2. In the first case, $\sum_{e \in P} y(e) \geq 1$ because $\sum_{e \in P} x(e) \geq 1$. In the second case, feasibility is ensured by the fact that $y(e_1) = y(e_2) = \alpha_i \geq \alpha_k \geq 1$.

Similarly, every s_2, t_2 path also either passes through N or uses e_1 or e_2 and so $\sum_{e \in P} y(e) \geq \alpha_i$ for all such paths. Hence y is feasible. □ The following claim proves the theorem.

Claim A.5 *We have $\sum_{e \in E[N']} c(e) y(e) = g_1' + \alpha_i g_2' = h_1' + \alpha_i h_2' = L_{N'}(\alpha_i)$.*

Proof: $\sum_{e \in E[N']} c(e) y(e) = \sum_{e \in E[N]} c(e) x(e) + 2C\alpha_i = L_N(\alpha_i) + 2C\alpha_i$. Also, $g_1' + \alpha_i g_2' = g_1 + \alpha_i g_2 + 2C\alpha_i$. The last equality in the claim follows from LP duality. □

We have shown that $(\alpha_1, \cdots, \alpha_k)$ are facets of N' as well. Now we show $(\zeta_k, \cdots, \zeta_1)$ are also facets. For a given j, let (g_1, g_2) and (h_1, h_2) be the feasible flows satisfying $f_1 + \alpha_j f_2 = L_N(\alpha_j)$. We now describe feasible flows (g_1', g_2') and (h_1', h_2') in N' and a solution $y(e)$ to $D_{N'}(\zeta_j)$ such that $g_1' + \zeta_j g_2' = h_1' + \zeta_j h_2' = \sum_{e \in E[N']} c(e) y(e)$. This will end the proof.

$$g'_1 = 2C - g_1,\; g'_2 = 2g_1 + g_2;\quad h'_1 = 2C - h_1,\; h'_2 = 2h_1 + h_2$$
$$\forall e \in E[N],\quad y(e) = (2\zeta_j - 1)x(e);\quad y(e_1) = y(e_2) = \zeta_j;\quad y(e_3) = y(e_4) = (1 - \zeta_j)$$

Claim A.6 *There exist feasible flows of value (g'_1, g'_2) and (h'_1, h'_2) in N'.*

Proof: Consider the flow of value g_1 from s_1 to t_1 through e_3, N, e_4 and a flow of value $C - g_1$ through the paths e_1, e_4 and e_3, e_2. This gives a flow of g'_1 from s_1 to t_1 in N'. Consider the flow of g_2 from s_2 to t_2 through N and a flow of value g_1 through both e_1 and e_2.This gives a flow of value g'_2 from s_2 to t_2 in N'. Note that this is feasible since (g_1, g_2) is feasible in N and C is a maximum $s_1 - t_1$ flow. □

Claim A.7 *The $y(e)$'s forms a feasible solution to the dual program $D_{N'}(\zeta_j)$.*

Proof: Consider a path P from s_1 to t_1 in N'. It either uses edges in N and e_3 and e_4 or it is one of the two: (e_1, e_4) or (e_3, e_2). The sum of y' of both the pairs in 1. So we may assume P is of the first kind. In this case,

$$y(P) = y(e_3) + y(e_4) + y(P \setminus \{e_3, e_4\}) = 2(1 - \zeta_j) + (2\zeta_j - 1)x(P \setminus \{e_3, e_4\}) \geq 1$$

where the last inequality follows from the fact that $P \setminus \{e_3, e_4\}$ is a path from s_1 to t_1 in N.

Now consider a path P from s_2 to t_2 in N'. The path either passes through N or consists of just e_1 or e_2. In the latter two cases we have by definition $y(P) = \zeta_j$. For every path passing through N we have

$$y(P) = (2\zeta_j - 1)x(P) \geq (2\zeta_j - 1)\alpha_j = \zeta_j$$

where the last equality follows from the definition of $\zeta_j = \frac{\alpha_j}{2\alpha_j - 1} \Rightarrow \alpha_j = \frac{\zeta_j}{2\zeta_j - 1}$. □

Claim A.8 *We have $\sum_{e \in E[N']} c(e)y(e) = g'_1 + \zeta_j g'_2 = h'_1 + \zeta_j h'_2 = L_{N'}(\zeta_j)$.*

Proof: Note that $\sum_{e \in E[N']} c(e)y(e) = (2\zeta_j - 1)\sum_{e \in E[N]} c(e)x(e) + 2C$. Now,

$$\begin{aligned} g'_1 + \zeta_j g'_2 &= (2C - g_1) + \zeta_j(2g_1 + g_2) = (2\zeta_j - 1)g_1 + \zeta_j g_2 + 2C \\ &= (2\zeta_j - 1)g_1 + (2\zeta_j - 1)\alpha_j g_2 + 2C = \sum_{e \in E[N']} c(e)y(e) \end{aligned}$$

where the second line follows from the fact that $\zeta_j = (2\zeta_j - 1)\alpha_j$. □

This proves Lemma A.2. □

Shifting Operation:
Given a network N, Figure 2 shows how to construct the stretched network N'. Suppose the profile of N was $(\alpha_1, \cdots, \alpha_k)$.

Lemma A.9 *The profile for N' is $(\alpha_1 + 1, \cdots, \alpha_k + 1)$.*

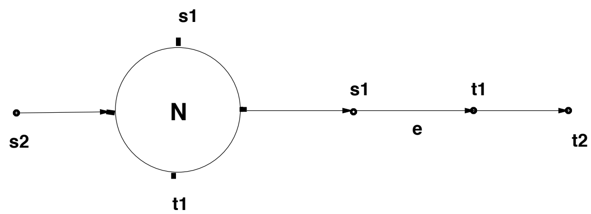

Figure 2: The network N' obtained from N. The edge e has a capacity D, where D is the maximum f_2 flow that is feasible in N.

Proof: For any $i = 1, \cdots, k$, let (g_1, g_2) and (h_1, h_2) be two feasible flows in N satisfying $f_1 + \alpha_i f_2 = L_N(\alpha_i)$. Let $x(e)$ be a feasible optimal dual solution to $D_N(\alpha_i)$. We construct (g_1', g_2') and (h_1', h_2') which are feasible flows in N' and a feasible dual solution $y(e)$ to $D_{N'}(\alpha_i + 1)$. We complete the proof by showing that $\sum_{e \in E[N']} c(e)y(e) = g_1' + (\alpha_i + 1)g_2' = h_1' + (\alpha_i + 1)h_2'$.

Let D be the max f_2 flow feasible in N. Note $D = Ł_N(\alpha_1)$

$$g_1' = g_1 + D - g_2,\ g_2' = g_2; \quad h_1' = h_1 + D - h_2,\ h_2' = h_2;$$
$$\forall e \in E[N],\ y(e) = x(e); \quad y(e) = 1$$

Claim A.10 *There exists feasible flows in N' of value (g_1', g_2') and (h_1', h_2') and $y(e)$'s form a feasible dual solution to $D_{N'}(\alpha_i + 1)$ whose value equals $g_1' + (\alpha_i + 1)g_2' = h_1' + (\alpha_i + 1)h_2'$.*

Proof: The flow of value g_1 through N and $D - g_2$ through the edge e is a feasible flow of value g_1' from s_1 to t_1. The flow of value g_2 via N and e is a valid g_2' flow.

The shortest s_1, t_1 paths P are either paths in N or the path $\{e\}$. In any case $y(P) \geq 1$. Any s_2, t_2 path has to be a path in N concatenated with e and thus $y(P) \geq \alpha_i + 1$.

Also, $\sum_{e \in E[N']} c(e)y(e) = \sum_{e \in E[N]} c(e)x(e) + D = L_N(\alpha_i) + D$. The proof completes by noticing that $g_1' + (\alpha_i + 1)g_2' = g_1 + D - g_2 + \alpha_i g_2 + g_2 = L_N(\alpha_i) + D$. The same is true for $h_1' + (\alpha_i + 1)h_2'$. $\square$ $\square$