# On the Implementation of Combinatorial Algorithms for the Linear Exchange Market

Kurt Mehlhorn(✉)

Max-Planck-Institute for Informatics, Saarbrücken, Germany
mehlhorn@mpi-inf.mpg.de

**Abstract.** Duan and Mehlhorn and Duan, Garg, and Mehlhorn presented polynomial time combinatorial algorithms [DM13,DGM15] for the computation of equilibrium prices in linear exchange markets. I am currently implementing these algorithms. I discuss the questions that I hope to answer through the implementation.

## 1 Introduction

In the linear exchange market model [Wal74] there are $n$ agents and $n$ goods; agent $i$ owns good $i$. Agents have preferences over goods. Let $u_{ij} \in \mathbb{N}_{\geq 0}$ be the utility of agent $i$ if all of good $j$ is allocated to him. Goods are divisible. At a certain vector $p = (p_1, \ldots, p_n)$ of prices, agents are only willing to spend money on goods that give them maximum utility per unit of money. Agents are sellers and buyers, i.e., if agent $i$ sells his good completely, he has a budget of $p_i$ units of money. The task is to compute prices at which the market clears, i.e., all goods are completely sold and all money is completely spent. Formally, we want to find a positive price vector $p$ and a nonnegative flow of money $f = (f_{ij})$ such that

$$\begin{aligned} p_i &= \textstyle\sum_j f_{ij} \quad \text{for all } i && \text{money is completely spent} \\ p_j &= \textstyle\sum_i f_{ij} \quad \text{for all } j && \text{goods are completely sold} \\ f_{ij} > 0 &\implies \tfrac{u_{ij}}{p_j} = \max_\ell \tfrac{u_{i\ell}}{p_\ell} && \text{agents are selfish} \end{aligned}$$

Fig. 1 shows an example.

The problem is solvable in polynomial time. Jain and Ye [Jai07,Ye07] gave algorithms based on the Ellipsoid and interior point method, respectively. Duan and Mehlhorn [DM13] provided a combinatorial algorithm which was recently improved by Duan, Garg, and Mehlhorn [DGM15]. We review the former algorithm in Sects. 2 and 3. We have recently started to implement the algorithm. In Sect. 4 we discuss the questions that we want to address through the implementation. A detailed description of the implementation is under preparation.

## 2 The Algorithm

This section and the next are based on [DM13]. Each agent only buys its favorite goods. Define the *bang per buck* of buyer $b_i$ to be $\alpha_i = \max_j u_{ij}/p_j$. For a price


C. Zaroliagis et al. (Eds.): Spirakis Festschrift, LNCS 9295, pp. 87–94, 2015.
DOI: 10.1007/978-3-319-24024-4_7

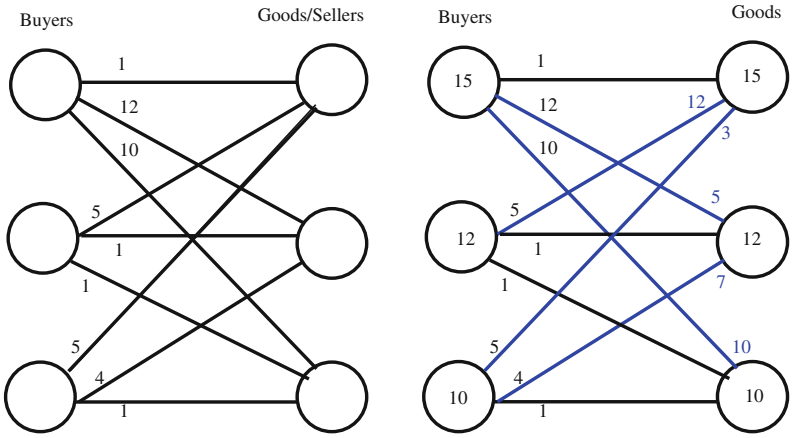


**Fig. 1.** The input is shown on the left. Each agent is shown twice, once in its role as seller or owner of a good and once in his role as buyer. The utility $u_{ij}$ is indicated on the edge from the $i$-th buyer to the $j$-th good. A solution is shown on the right. The prices of the goods (= budgets of the buyers) are shown inside the nodes. The bang-for-buck edges and the money flow are shown in blue.

vector $p$, the *equality network* $N_p$ is a flow network with vertex set $\{s,t\} \cup B \cup C$, where $s$ is a source node, $t$ is a sink node, $B = \{b_1, \ldots, b_n\}$ is the set of buyers, and $C = \{c_1, \ldots, c_n\}$ is the set of goods, and the following edge set:

- An edge $(s, b_i)$ with capacity $p_i$ for each $b_i \in B$.
- An edge $(c_i, t)$ with capacity $p_i$ for each $c_i \in C$.
- An edge $(b_i, c_j)$ with infinite capacity whenever $u_{ij}/p_j = \alpha_i$. We use $E_p$ to denote these edges.

Our task is to find a positive price vector $p$ such that there is a flow in which all edges from $s$ and to $t$ are saturated. When this is satisfied, all goods are sold and all of the money earned by each agent is spent on goods of maximum utility per unit of money. With respect to a flow $f$, define the surplus $r(b_i)$ of a buyer $i$ as $r(b_i) = p_i - \sum_j f_{ij}$, where $f_{ij}$ is the amount of flow on the edge $(b_i, c_j)$, and define the surplus $r(c_j)$ of a good $j$ as $r(c_j) = p_j - \sum_i f_{ij}$, Define the surplus vector of buyers to be $r = (r(b_1), r(b_2), ..., r(b_n))$. Also, define the total surplus to be $|r| = \sum_i r(b_i)$, which is also $\sum_j r(c_j)$ since the total capacity from $s$ and to $t$ are both equal to $\sum_i p_i$. For convenience, we denote the surplus vector of flow $f'$ by $r'$. In the network corresponding to market clearing prices, the total surplus of a maximum flow is zero.

A maximum flow $f$ in $N_p$ is *balanced* if it minimizes the 2-norm $\|r\|_2$ of the surplus vector of the buyers among all maximum flows. Balanced flows were introduced in [DPSV08] and shown to be computable by $n$ maxflow computations in $N_p$. The equality graph may have as many as $\Theta(n^2)$ edges and hence, assuming the use of an $O(nm + n^2 \log n)$ maxflow algorithm, a balanced flow can

Set $\epsilon = 1/(8n^{4n}U^{3n})$, where $U = \max_{ij} u_{ij}$;
Set $p_i = 1$ for all $i$ and set $f$ to a balanced flow in $N_p$;
**Repeat**
  Sort the buyers by their surpluses in decreasing order: $b_1, b_2, ..., b_n$;
  Find the smallest $\ell$ for which $r(b_\ell)/r(b_{\ell+1}) > 1 + 1/n$, and
    let $\ell = n$ when there is no such $\ell$;
  Let $S = \{b_1, \ldots, b_\ell\}$;
  Compute $x = \min(x_{eq}, x_{23}, x_{24}, x_2)$.
  Multiply prices of goods in $\Gamma(S)$ and flows into these goods by $x$.
  Extend $f$ to a balanced flow (goods of surplus zero must keep surplus zero);
**Until** $|r(B)| < \epsilon$;
Extract a linear system from $N_p$ and compute equilibrium prices from it.

**Fig. 2.** The algorithm

be computed with $O(n^4)$ arithmetic operations. The algorithm for computing a balanced flow is based on the following characterization of balanced flows.

**Lemma 1 [DPSV08].** *A maximum flow $f$ is balanced if for any two edges $(b_i, c_k)$ and $(b_j, c_k)$ on the equality graph with $f_{ik} > 0$, either $r(b_i) = r(b_j)$ or $(r(b_i) > r(b_j)$ and $f_{jk} = 0)$.*

The algorithm is shown in Fig. 2. It starts with all prices $p_i$ equal to one and a balanced flow $f$ in $N_p$. It works in phases. In each phase, we first number the buyers in order of decreasing surpluses: $b_1, \ldots, b_n$. Let $\ell$ be minimal such that $r(b_\ell)$ is by a factor of $1 + 1/n$ larger than $r(b_{\ell+1})$; $\ell = n$ if there is no such $\ell$. By this choice of $\ell$, $r(b_\ell) \geq |r(B)|/(e \cdot n)$ and $r(b_i) \leq e \cdot r(b_\ell)$ for all $i$ is guaranteed.[1] Let $S = \{b_1, \ldots, b_\ell\}$ and let $\Gamma(S) = \{c \in C \mid (b, c) \in E_p \text{for some} b \in S\}$ be the goods that are adjacent to a buyer in $S$ in the equality graph. Since every buyer has at least one incident edge in the equality graph, $\Gamma(S)$ is non-empty. All flow from buyers in $S$ goes to goods in $\Gamma(S)$ and all buyers in $S$ have surplus. Thus the goods in $\Gamma(S)$ have no surplus, since the current flow is maximum, and the demand for them at the current prices exceeds their supply.

Let $\bar{S} = B \setminus S$. Since the flow is balanced and the buyers in $S$ have larger surplus than the buyers in $\bar{S}$, there is no flow from $\bar{S}$ to $\Gamma(S)$. We raise the prices of the goods in $\Gamma(S)$ and the flow on the edges incident to them by a common factor $x > 1$. We also increase the flow from $s$ to buyers in $S$ such that flow conservation holds. This give us a new price vector $p'$ and a new flow $f'$. Observe that the surpluses of the goods in $\Gamma(S)$ stay zero. Formally,

[1] Clearly $b_1 \geq |r(B)|/n$. Also, $r(b_j)/r(b_{j+1}) \leq 1 + 1/n$ for $j < \ell$, and hence $r(b_\ell) \geq r(b_1)/(1 + 1/n)^{-n} \geq r(b_1)/e \geq |r(B)|/(e \cdot n)$.

$$p'_j = \begin{cases} x \cdot p_j & \text{if } c_j \in \Gamma(S); \\ p_j & \text{if } c_j \notin \Gamma(S). \end{cases} \quad (1) \qquad f'_{ij} = \begin{cases} x \cdot f_{ij} & \text{if } c_j \in \Gamma(S); \\ f_{ij} & \text{if } c_j \notin \Gamma(S). \end{cases} \quad (2)$$

The changes on the edges incident to $s$ and $t$ are implied by flow conservation.

The change of prices and flows affects the surpluses of the buyers, some go up and some go down.

**Lemma 2 [DM13].** *Given a balanced flow $f$ in $N_p$, a set $S$ of buyers such that all goods in $\Gamma(S)$ are completely sold and there is no flow from $\bar{S}$ to $\Gamma(S)$, and a sufficiently small parameter $x > 1$, the flow $f'$ defined in (2) is a feasible flow in the equality network with respect to the prices in (1). The surplus of each good remains unchanged, and the surpluses of the buyers become:*

$$r'(b_i) = \begin{cases} x \cdot r(b_i) & \text{if } b_i \in S, c_i \in \Gamma(S) \text{ (type 1 buyer)}; \\ (1-x)p_i + x \cdot r(b_i) & \text{if } b_i \in S, c_i \notin \Gamma(S) \text{ (type 2 buyer)}; \\ (x-1)p_i + r(b_i) & \text{if } b_i \notin S, c_i \in \Gamma(S) \text{ (type 3 buyer)}; \\ r(b_i) & \text{if } b_i \notin S, c_i \notin \Gamma(S) \text{ (type 4 buyer)}. \end{cases}$$

*Proof.* See [DM13]. For the definition of the factor $x$, we perform the following thought experiment. We increase the prices of the goods in $\Gamma(S)$ and the flow on the edges incident to them continuously by a common factor $x$ until one of three events happens: (1) a new edge enters the equality graph or (2) the surplus of a type 2 buyer and a type 3 or 4 buyer become equal or (3) the surplus of a type 2 buyer becomes zero. The third event can only happen if $S = B$ and hence there are no type 3 and type 4 buyers.

The increase of prices of goods in $\Gamma(S)$ makes the goods in $C \setminus \Gamma(S)$ more attractive to the buyers in $S$ and hence an equality edge connecting a buyer in $S$ with a good in $C \setminus \Gamma(S)$ may arise. This will happen at $x = x_{eq}(S)$, where

$$x_{eq}(S) = \min \left\{ \frac{u_{ij}}{p_j} \cdot \frac{p_k}{u_{ik}} \mid b_i \in S, (b_i, c_j) \in E_p, c_k \notin \Gamma(S) \right\}.$$

When we increase the prices of the goods in $\Gamma(S)$ by a common factor $x \leq x_{eq}(S)$, the equality edges in $(S \times \Gamma(S)) \cup (\bar{S} \times (C \setminus \Gamma(S)))$ will remain in the network. Equality edges in $\bar{S} \times \Gamma(S)$ will disappear, but they carry no flow and hence may disappear.

The surplus of type 1 and 3 buyers increases, the surplus of type 2 buyers decreases, and the surplus of type 4 buyers does not change. Since the total surplus does not change (recall that the surpluses of the goods are not affected by the price update), the decrease in surplus of the type 2 buyers is equal to the increase in surplus of the type 1 and 3 buyers. In particular, there are type 2 buyers. We define quantities $x_{23}(S)$ and $x_{24}(S)$ at which the surplus of a type 2 and type 3 buyer, respectively type 4 buyer, becomes equal, and a quantity $x_2$ at which the surplus of a type 2 buyer becomes zero.

$$x_{23}(S) = \min\left\{\frac{p_i + p_j - r(b_j)}{p_i + p_j - r(b_i)} b_i \text{ is type 2 and } b_j \text{ is type 3 buyer}\right\},$$
$$x_{24}(S) = \min\left\{\frac{p_i - r(b_j)}{p_i - r(b_i)} b_i \text{ is type 2 and } b_j \text{ is type 4 buyer}\right\},$$
$$x_2(S) = \min\left\{\frac{p_i}{p_i - r(b_i)}\ b_i \text{ is type 2 buyer}\right\}.$$

The quantity $x_2(S)$ is only relevant, if $S = B$. It guarantees that surpluses of buyers stay nonnegative.

**Lemma 3 [DM13].** *With $x = \min(x_{eq}(S), x_{23}(S), x_{24}(S), x_2(S))$ and $S$ as defined in the algorithm, $f'$ is a feasible flow in $N_{p'}$.*

*Proof* Obvious. We complete the phase by extending $f'$ to a balanced flow. In this step, we make sure that goods with surplus zero keep surplus zero. This ends the description of the algorithm. Correctness follows from the following lemma.

**Lemma 4 [DM13].** *Once the surplus of a good becomes zero, it stays zero. As long as a good has non-zero surplus, its price stays at one.*

## 3 A Glimpse of the Analysis

The analysis uses two potential functions, namely the product $P = \prod_i p_i$ of all prices and the 2-norm $\|r(B)\|_2$ of the surplus vector of the buyers.

**Lemma 5 [DM13].** *In the course of the algorithm, all prices stay bounded by $(nU)^n$.*

**Lemma 6 [DM13].** *For a phase $h$, let $x_h > 1$ be the factor by which the prices in $\Gamma(S)$ are increased. Then*

$$\prod_h x_h \leq (nU)^{n^2}.$$

Let $x_{\max} := 1 + \frac{1}{48e^2n^3}$. Call a phase $h$ an $x_{\max}$-phase if $x_h \geq x_{\max}$ and call it a balancing phase otherwise.

**Lemma 7 [DM13].** *The number of $x_{\max}$-phases is $O(n^5 \log(nU))$.*

*Proof.* Let $T$ be the number of $x_{\max}$-phases. In an $x_{\max}$-phase, the product $P$ of the prices grows by at least a factor $x_{\max}$. Therefore $x_{\max}^T \leq (nU)^{n^2}$.

The 2-norm of the surplus vector is used to bound the number of balancing phases. The next lemma justifies the name.

**Lemma 8 [DM13].** *In a phase $h$ with $x_h < x_{\max}$, the 2-norm of the surplus vector of the buyers is reduced by a factor $1 - O(1/n^3)$.*

**Lemma 9** [DM13]. *Over all $x_{\max}$-phases, the 2-norm of the surplus vector of the buyers increases by at most a multiplicative factor $(nU)^{n^2}$.*

**Lemma 10** [DM13]. *The number of balancing phases is $O(n^5 \log(nU))$.*

*Proof.* Let $T$ be the number of balancing phases. The initial 2-norm of the surplus vector is at most $\sqrt{n}$. The total multiplicative increase is at most $(nU)^{n^2}$ and the total multiplicative decrease in balancing phases is at least $(1 - O(1/n^3))^T$. The algorithm terminates once the 1-norm of the surplus vector is less than $\epsilon$. This is guaranteed if the 2-norm is less than $\epsilon/\sqrt{n}$. The bound follows.

**Theorem 1** [DM13]. *The total number of arithmetic operations required by the algorithm is $O(n^9 \log(nU))$.*

*Polynomial Time:* Since we assume utilities to be integers and the algorithm uses only the basic arithmetic operations, the computation stays within the rationals. However, it is not clear whether the size of the rationals stays polynomially bounded. It is conceivable that the size of the rationals doubles in each phase.

Duan and Mehlhorn guarantee polynomial time as follows. Firstly, they approximate utilities $u_{ij}$ by powers of $1 + 1/L$, where $L = 16n^5(nU)^n/\epsilon = 126n^{5n+5}U^{4n}$. Secondly, they observe that it suffices to approximate $x_{23}$, $x_{24}$ and $x_2$ by a nearby power of $1 + 1/L$ and that only $x_{eq}$ needs to be computed exactly. As a consequence, all prices are powers of $1 + 1/L$. Since prices are bounded by $(nU)^n$, the exponents in the representations are between 0 and $\log_{1+1/L}(nU)^n = O(nL \log(nU)$. The bitlength of the exponents is $O(n \log(nU))$.

**Theorem 2** [DM13]. *The total number of arithmetic operations required by the algorithm is $O(n^9 \log(nU))$. Arithmetic on integers with $O(n \log(nU))$ bits suffices.*

## 4 Questions

Through the implementation, we address the following questions:

1. Do the rationals really explode? Or does it suffice to keep them normalized, i.e., to keep numerators and denominators relatively prime? For Gaussian elimination over the rationals it is known [Edm67] that keeping the rationals normalized suffices to control their size.
2. The known algorithm for computing a balanced flow requires up to $n$ maxflow computations in a graph with $2n$ nodes and $O(n^2)$ edges.
   (a) Cycles in the equality graph can only arise if there are dependencies between the utilities. Let $b_{i_0}, c_{j_0}, b_{i_1}, c_{j_1}, \ldots, b_{i_{k-1}}, c_{j_{k-1}}, b_{i_0}$ be a cycle in the equality graph with respect to a price vector $p$. Then

$$\frac{u_{i_\ell, j_{\ell-1}}}{p_{j_{\ell-1}}} = \frac{u_{i_\ell, j_\ell}}{p_{j_\ell}}$$

for all $\ell$, since $b_{i_\ell}$ is connected to $c_{j_{\ell-1}}$ and $c_{j_\ell}$ in the equality graph (interpret $-1$ as $k-1$) and hence

$$\prod_{0\le\ell\le k-1} u_{i_\ell,j_{\ell-1}} = \prod_{0\le\ell\le k-1} u_{i_\ell,j_\ell}.$$

If no such dependency exists, the equality graph $E_p$ is a tree and contains at most $2n-1$ edges. Thus the maxflow computations would be in graphs with only a linear number of edges and special structure. Perturbation of the utilities can guarantee that there are no dependencies. What is the arithmetic cost of perturbation in this context? Does the rounding of the utilities required to make the algorithm polynomial time counteract perturbation and introduce dependencies?

(b) The algorithm for computing a balanced flow works recursively. The divide step answers the question whether all surpluses can be made equal. Note that the average surplus of a buyer is $r_{ave} := (\sum_{i\in B}(p_i - \sum_j f_{ij}))/n$. We set the capacity of the edge from $s$ to $b_i$ to $p_i - r_{ave}$ and recompute the maximum flow. If the entire flow can still be routed, all surpluses are equal to $r_{ave}$ in a balanced flow and we are done. Otherwise, let $S$ be the set of buyers and goods reachable from $s$ in the residual graph and let $T$ be their complement. The nodes in $S \cap B$ have surplus at least $r_{ave}$ in a balanced flow and the nodes in $B \cap T$ have surplus at most $r_{ave}$. One deletes all edges from buyers in $T$ to goods in $S$ from the equality graph and recurses on the graphs $s \cup S \cup t$ and $s \cup T \cup t$. Actually, the two recursive calls can be combined into one as there are no edges connecting nodes in $S$ and nodes in $T$. In this way, the number of recursive calls is equal to the recursion depth. Can one control the recursion depth, by using a value different from $r_{ave}$ in the divide step?
Note that the networks for the recursive calls are obtained by deleting some edges and changing the capacities of the edges incident to $s$. Does one have to start the maxflow computations from scratch or can one reuse the results of earlier computations.

(c) Can one use parametric maxflow [GGT89] to speed up the computation?

(d) It seems that one can do with a weaker version of balanced flows, namely $1+1/(cn)$-balanced flows where $c$ is a small constant. A maximum flow is $1+1/(cn)$-balanced if for any two edges $(b_i, c_k)$ and $(b_j, c_k)$ in the equality graph with $f_{ik} > 0$, we have either $r(b_j) \ge r(b_i)/(1+1/cn))$ or $f_{jk} = 0$.

3. What is the behavior of the algorithm on random inputs? Are their inputs that force the algorithm into its worst-case running time or close to the worst-case running time.
4. Are balanced flows really needed? Garg, Duan, and Mehlhorn [DGM15] have recently shown that the use of balanced flows can be avoided. This reduces the complexity of a phase to $O(n^2)$. Does this theoretical improvement show in the implementation?
5. The main loop terminates when the 1-norm of the surplus vector of the buyers is less than $\epsilon$. At this point, one can extract a linear system from the equality graph $E_p$. The equilibrium prices are the solution to this linear system.

Of course, one can extract a linear system from $E_p$ at any time during the execution and compute prices from it. Since it is easily checked whether a set of prices is a set of equilibrium prices (one maxflow computation), it makes sense to extract earlier. If the cost of extracting and solving the system is $C$, one should extract after spending cost $O(C)$ in the main loop. In this way the extraction attempts can be amortized over the cost of the main loop.

6. What problem size can be solved with an $O(n^{10})$ algorithm? In the worst case? On average? Are economists interested in exact solutions to problems of this size?

## References

[DGM15] Duan, R., Garg, J., Mehlhorn, K.: A improved combinatorial algorithm for the linear arrow-debreu marketTODO (2015). Forthcoming

[DM13] Duan, R., Mehlhorn, K.: A combinatorial polynomial algorithm for the linear arrow-debreu market. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) ICALP 2013, Part I. LNCS, vol. 7965, pp. 425–436. Springer, Heidelberg (2013)

[DPSV08] Devanur, N.R., Papadimitriou, C.H., Saberi, A., Vazirani, V.V.: Market equilibrium via a primal-dual algorithm for a convex program. J. ACM **55**(5), 22:1–22:18 (2008)

[Edm67] Edmonds, J.: Systems of distinct representatives and linear algebra. J. Res. Nat. Bur. Stan.(B) **71**, 241–245 (1967)

[GGT89] Gallo, G., Grigoriadis, M.D., Tarjan, R.E.: A fast parametric maximum flow algorithm and applications. SIAM J. Comput. **18**, 30–55 (1989)

[Jai07] Jain, K.: A polynomial time algorithm for computing an Arrow-Debreu market equilibrium for linear utilities. SIAM J. Comput. **37**(1), 303–318 (2007)

[Wal74] Walras, L.: Elements of Pure Economics, or the Theory of Social Wealth (1874)

[Ye07] Ye, Y.: A path to the Arrow-Debreu competitive market equilibrium. Math. Program. **111**(1), 315–348 (2007)