

A Note on Balanced Flows in Equality Networks

Kamiyama, Naoyuki
Institute of Mathematics for Industry, Kyushu University

<https://hdl.handle.net/2324/4479690>

出版情報 : Information Processing Letters. 145, pp.74-76, 2019-05. Elsevier
バージョン :
権利関係 :



A Note on Balanced Flows in Equality Networks

Naoyuki Kamiyama

*Institute of Mathematics for Industry, Kyushu University
JST, PRESTO*

Abstract

A balanced flow in an equality network is an important tool in the problem of finding the equilibrium prices in the linear Fisher market and the linear Arrow–Debreu market. In this paper, we prove that the problem of finding a balanced flow in an equality network can be reduced to the problem of finding a lexicographically optimal flow in a slightly modified network.

Keywords: balanced flow, lexicographically optimal flow, equilibrium

1. Introduction

A balanced flow in an equality network introduced by Devanur, Papadimitriou, Saberi, and Vazirani [1] is an important tool in the problem of finding the equilibrium prices in the linear Fisher market [1] and the linear Arrow–Debreu market [2, 3] (see also [4, 5, 6, 7]). An equality network is a network where we have the set of buyers adjacent to the source vertex and the set of goods adjacent to the sink vertex, and there are edges between the set of buyers and the set of goods. It is known [1] that a balanced flow can be found with $O(b)$ maximum flow computations, where b is the number of buyers. Darwish and Mehlhorn [8] proposed a faster algorithm for finding a balanced flow in an equality network by using parametric maximum flow [9]. The time complexity of the algorithm of Darwish and Mehlhorn [8] is the same as the time complexity of the algorithm of [10] for the maximum flow problem.

In this paper, we prove that the problem of finding a balanced flow in an equality network can be reduced to the problem of finding a lexicographically optimal flow [11] in a slightly modified network. (Precisely speaking, we consider a slightly generalized setting.) It is known [9] that we can find a lexicographically optimal flow in the same time complexity as the algorithm of [10] for the maximum flow problem. Notice that this time complexity matches the result of Darwish and Mehlhorn [8].

Email address: kamiyama@imi.kyushu-u.ac.jp (Naoyuki Kamiyama)

2. Preliminaries

We denote by \mathbb{R}_+ , \mathbb{Q}_+ , and \mathbb{Q}_{++} the sets of non-negative real numbers, non-negative rational numbers, and positive rational numbers, respectively. For each finite set U , each function $\xi: U \rightarrow \mathbb{R}$, and each subset X of U , we define $\xi(X) := \sum_{u \in X} \xi(u)$. Assume that we are given ordered sequences $x = (x_1, x_2, \dots, x_\ell)$ and $y = (y_1, y_2, \dots, y_\ell)$ of non-negative real numbers. Then we write $x >_{\text{lex}} y$, if there exists an integer i in $\{1, 2, \dots, \ell\}$ such that (i) for every integer i in \mathbb{Z}_+ such that $1 \leq j < i$, $x_j = y_j$, and (ii) $x_i > y_i$.

For each finite directed graph D , we denote by $V(D)$ and $A(D)$ the vertex set and the arc set of D , respectively. For each finite directed graph D and each vertex v in $V(D)$, we denote by $\delta_D^+(v)$ (resp., $\delta_D^-(v)$) the set of arcs $a = (u, w)$ in $A(D)$ such that $u = v$ (resp., $w = v$). In this paper, we assume that for each finite directed graph D , we are given a subset $S(D)$ of $V(D)$ as the set of sources of D , and some vertex $t(D)$ in $V(D) \setminus S(D)$ is designated as a single sink of D . Furthermore, we assume that for every finite directed graph D , $\delta_D^-(v) = \emptyset$ for every vertex v in $S(D)$, and $\delta_D^+(t(D)) = \emptyset$.

In this paper, we define a *network* N as a pair of a finite directed graph D and a capacity function $c: A(D) \rightarrow \mathbb{Q}_+$. We denote by $N = (D, c)$ this network. For each network $N = (D, c)$, a function $f: A(D) \rightarrow \mathbb{R}_+$ is called a *feasible flow* in N , if the following conditions are satisfied.

- For every arc a in $A(D)$, $f(a) \leq c(a)$.
- For every vertex v in $V(D) \setminus (S(D) \cup \{t(D)\})$, $f(\delta_D^-(v)) = f(\delta_D^+(v))$.

Assume that we are given a network $N = (D, c)$. For each feasible flow f in N , we define the function $\text{val}_f: S(D) \rightarrow \mathbb{R}_+$ by $\text{val}_f(v) := f(\delta_D^+(v))$. A feasible flow f in N is called a *maximum flow* in N , if

$$\text{val}_f(S(D)) \geq \text{val}_g(S(D))$$

for every feasible flow g in N .

3. Balanced Flows

In this section, we formally define the *balanced flow problem*. An instance of the balanced flow problem consists of a network $N = (D, c)$, a weight function $\omega: S(D) \rightarrow \mathbb{Q}_{++}$, and a source capacity function $e: S(D) \rightarrow \mathbb{Q}_+$. We assume that the following conditions are satisfied.

- D does not have parallel arcs and self-loops.
- There does not exist an arc from a vertex in $S(D)$ to $t(D)$.
- For every vertex v in $S(D)$, $\delta_D^+(v)$ consists of a single arc a_v .
- For every vertex v in $S(D)$, $c(a_v) = e(v)$.

- $\omega(v) \geq 1$ for every vertex v in $S(D)$.

We denote by $I = (N = (D, c), \omega, e)$ this instance.

Assume that we are given an instance $I = (N = (D, c), \omega, e)$ of the balanced flow problem. Then the goal of the balance flow problem is to find a maximum flow f in N such that

$$\sqrt{\sum_{v \in S(D)} \frac{(e(v) - \text{val}_f(v))^2}{\omega(v)}} \leq \sqrt{\sum_{v \in S(D)} \frac{(e(v) - \text{val}_g(v))^2}{\omega(v)}} \quad (1)$$

55 for every maximum flow g in N .

Darwish and Mehlhorn [8] considered this problem in an equality network (i.e., the structure of D is restricted and $\omega(v) = 1$ for every vertex v in $S(D)$), and proved that the balanced flow problem can be solved in $O(nm \log(n^2/m))$ time by using the parametric maximum flow algorithm of [9], where we define
60 $n := |V(D)|$ and $m := |A(D)|$.

4. Lexicographically Optimal Flows

In this section, we formally define the *lexicographically optimal flow problem*. An instance of the lexicographically optimal flow problem consists of a network $N = (D, c)$ and a weight function $\omega: S(D) \rightarrow \mathbb{Q}_{++}$. We denote by $I = (N =$
65 $(D, c), \omega)$ this instance.

Assume that we are given an instance $I = (N = (D, c), \omega)$ of the lexicographically optimal flow problem. For each feasible flow f in N , we define

$$\mathbf{v}_I(f) := \left(\frac{\text{val}_f(s_1)}{\omega(s_1)}, \frac{\text{val}_f(s_2)}{\omega(s_2)}, \dots, \frac{\text{val}_f(s_k)}{\omega(s_k)} \right),$$

where we assume that $S(D) = \{s_1, s_2, \dots, s_k\}$ and

$$\frac{\text{val}_f(s_1)}{\omega(s_1)} \leq \frac{\text{val}_f(s_2)}{\omega(s_2)} \leq \dots \leq \frac{\text{val}_f(s_k)}{\omega(s_k)}.$$

Notice that the order s_1, s_2, \dots, s_k depends on f . A feasible flow f in N is called a *lexicographically optimal flow in N* , if $\mathbf{v}_I(f) >_{\text{lex}} \mathbf{v}_I(g)$ or $\mathbf{v}_I(f) = \mathbf{v}_I(g)$ for every maximum flow g in N . Then the goal of the lexicographically optimal flow problem is to find a lexicographically optimal flow in N .

70 It is known [9, Section 4.1] that the lexicographically optimal flow problem can be solved in $O(nm \log(n^2/m))$ time by using the parametric maximum flow algorithm of [9], where we define $n := |V(D)|$ and $m := |A(D)|$.

Furthermore, it is known [12, Theorem 3.3] that for every instance $I = (N = (D, c), \omega)$ of the lexicographically optimal flow problem, I is equivalent to the problem of finding a maximum flow f in N such that

$$\frac{1}{2} \sum_{v \in S(D)} \frac{\text{val}_f(v)^2}{\omega(v)} \leq \frac{1}{2} \sum_{v \in S(D)} \frac{\text{val}_g(v)^2}{\omega(v)}$$

for every maximum flow g in N .

5. Main Result

Assume that we are given an instance $I = (N = (D, c), \omega, e)$ of the balanced flow problem. Define

$$M := \max\{e(v) \mid v \in S(D)\}.$$

75 Notice that for every maximum flow f in N , $\text{val}_f(S(D))$ is the same. We denote by B this value.

Define the instance $I' = (N' = (D', c'), \omega')$ of the lexicographically optimal flow problem as follows.

- $V(D') := V(D)$, $S(D') := S(D)$, and $t(D') := t(D)$.
- 80 • $A(D') := A(D) \cup \{\bar{a}_v = (v, t(D)) \mid v \in S(D)\}$, where \bar{a}_v is a new arc that is not contained in $A(D)$.
- $c'(a) := c(a)$ for each arc a in $A(D)$, and $c'(\bar{a}_v) := \omega(v) \cdot M - e(v)$ for each vertex v in $S(D)$. (Notice that for every vertex v in $S(D)$, since $\omega(v) \geq 1$, we have $c'(\bar{a}_v) \geq 0$.)
- 85 • $\omega' := \omega$.

Theorem 1. *Assume that f' is an optimal solution of I' . Define the function $f: A(D) \rightarrow \mathbb{R}_+$ by $f(a) := f'(a)$. Then f is an optimal solution of I .*

Proof. It is not difficult to see that f is a maximum flow in N . We prove that (1) holds for every maximum flow g in N . Define the function $g': A(D') \rightarrow \mathbb{R}_+$ by $g'(a) := g(a)$ for each arc a in $A(D)$ and $g'(\bar{a}_v) := c'(\bar{a}_v)$ for each vertex v in $S(D)$. It is not difficult to see that g' is a maximum flow in N' . Since f' is a maximum flow in N' , we have $f'(\bar{a}_v) = c'(\bar{a}_v) = \omega(v) \cdot M - e(v)$ for every vertex v in $S(D')$. Thus, we have

$$\begin{aligned} \sum_{v \in S(D')} \frac{\text{val}_{f'}(v)^2}{\omega'(v)} &= \sum_{v \in S(D')} \frac{(f'(a_v) + f'(\bar{a}_v))^2}{\omega'(v)} \\ &= \sum_{v \in S(D)} \frac{(\text{val}_f(v) + \omega(v) \cdot M - e(v))^2}{\omega(v)} \\ &= \sum_{v \in S(D)} \frac{(\text{val}_f(v) - e(v))^2}{\omega(v)} + \sum_{v \in S(D)} \left(2M \cdot (\text{val}_f(v) - e(v)) + \omega(v) \cdot M^2 \right) \\ &= \sum_{v \in S(D)} \frac{(e(v) - \text{val}_f(v))^2}{\omega(v)} + 2M \cdot (B - e(S(D))) + \omega(S(D)) \cdot M^2. \end{aligned} \tag{2}$$

Since g is a maximum flow in N , in the same way, we can prove that

$$\begin{aligned} \sum_{v \in S(D')} \frac{\text{val}_{g'}(v)^2}{\omega'(v)} \\ &= \sum_{v \in S(D)} \frac{(e(v) - \text{val}_g(v))^2}{\omega(v)} + 2M \cdot (B - e(S(D))) + \omega(S(D)) \cdot M^2. \end{aligned} \tag{3}$$

Since f' is an optimal solution of I' and g' is a maximum flow in N' ,

$$\frac{1}{2} \sum_{v \in S(D')} \frac{\text{val}_{f'}(v)^2}{\omega'(v)} \leq \frac{1}{2} \sum_{v \in S(D')} \frac{\text{val}_{g'}(v)^2}{\omega'(v)}. \quad (4)$$

Since (2), (3), and (4) imply that

$$\sum_{v \in S(D)} \frac{(e(v) - \text{val}_f(v))^2}{\omega(v)} \leq \sum_{v \in S(D)} \frac{(e(v) - \text{val}_g(v))^2}{\omega(v)},$$

we have

$$\sqrt{\sum_{v \in S(D)} \frac{(e(v) - \text{val}_f(v))^2}{\omega(v)}} \leq \sqrt{\sum_{v \in S(D)} \frac{(e(v) - \text{val}_g(v))^2}{\omega(v)}}.$$

This completes the proof. \square

Assume that $|V(D)| = n$ and $|A(D)| = m$. Since $|V(D')| = n$ and $|A(D')| \leq m + n$, Theorem 1 implies that the balanced flow problem can be solved in $O(nm \log(n^2/m))$ time by using the parametric maximum flow algorithm of [9]. (Without loss of generality, we assume that for every vertex v in $V(D)$, we have $|\delta_D^+(v)| + |\delta_D^-(v)| \geq 1$. This implies that $n \leq 2m$.)

Acknowledgements. This research was supported by JST PRESTO Grant Number JPMJPR1753, Japan.

References

- [1] N. R. Devanur, C. H. Papadimitriou, A. Saberi, V. V. Vazirani, Market equilibrium via a primal–dual algorithm for a convex program, *Journal of the ACM* 55 (5) (2008) 22:1–22:18.
- [2] R. Duan, K. Mehlhorn, A combinatorial polynomial algorithm for the linear Arrow–Debreu market, *Information and Computation* 243 (2015) 112–132.
- [3] R. Duan, J. Garg, K. Mehlhorn, An improved combinatorial polynomial algorithm for the linear Arrow–Debreu market, in: *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2016, pp. 90–106.
- [4] X. Bei, J. Garg, M. Hoefer, K. Mehlhorn, Computing equilibria in markets with budget-additive utilities, in: *Proceedings of the 24th Annual European Symposium on Algorithms*, Vol. 57 of *Leibniz International Proceedings in Informatics*, 2016, pp. 8:1–8:14.
- [5] X. Bei, J. Garg, M. Hoefer, Ascending-price algorithms for unknown markets, in: *Proceedings of the 2016 ACM Conference on Economics and Computation*, 2016, p. 699.

- [6] V. V. Vazirani, The notion of a rational convex program, and an algorithm for the Arrow-Debreu Nash bargaining game, *Journal of the ACM* 59 (2) (2012) 7:1–7:36.
- 115 [7] V. V. Vazirani, Spending constraint utilities with applications to the ad-words market, *Mathematics of Operations Research* 35 (2) (2010) 458–478.
- [8] O. Darwish, K. Mehlhorn, Improved balanced flow computation using parametric flow, *Information Processing Letters* 116 (9) (2016) 560–563.
- 120 [9] G. Gallo, M. D. Grigoriadis, R. E. Tarjan, A fast parametric maximum flow algorithm and applications, *SIAM Journal on Computing* 18 (1) (1989) 30–55.
- [10] A. V. Goldberg, R. E. Tarjan, A new approach to the maximum-flow problem, *Journal of the ACM* 35 (4) (1988) 921–940.
- 125 [11] N. Megiddo, Optimal flows in networks with multiple sources and sinks, *Mathematical Programming* 7 (1) (1974) 97–107.
- [12] S. Fujishige, Lexicographically optimal base of a polymatroid with respect to a weight vector, *Mathematics of Operations Research* 5 (2) (1980) 186–196.