# Concave Generalized Flows with Applications to Market Equilibria

*Preliminary draft, please do not distribute!*

László A. Végh *

July 23, 2011

**Abstract**

We consider a nonlinear extension of the generalized network flow model, with the flow leaving an arc being an increasing concave function of the flow entering it, as proposed by Truemper [37] and Shigeno [33]. We give the first polynomial time combinatorial algorithm for solving corresponding optimization problems, finding a $\varepsilon$-approximate solution in $O(m(m+\log n)\log(MUm/\varepsilon))$ arithmetic operations and value oracle queries, where $M$ and $U$ are upper bounds on simple parameters. For (linear) generalized flows, our algorithm can be seen as a variant of the highest-gain augmenting path algorithm by Goldfarb, Jin and Orlin [15].

We show that this general convex programming model serves as a common framework for several market equilibrium problems, including the linear Fisher market model and its various extensions. Our result immediately enables us to extend these market models to more general settings and to solve some open problems in the literature.

## 1 Introduction

A classical extension of network flows is the *generalized network flow model*, with a gain factor $\gamma_e > 0$ associated with each arc $e$ so that if $\alpha$ units of flow enter arc $e$, then $\gamma_e\alpha$ units leave it. Since first studied in the sixties by Dantzig [5] and Jewell [20], the problem has found many applications including financial analysis, transportation, managment sciences, etc.; see [2, Chapter 15].

In this paper, we consider a nonlinear extension, *concave generalized flows*, studied by Truemper [37] in 1978, and by Shigeno [33] in 2006. For each arc $e$ we are given a concave, monotone increasing function $\Gamma_e$ such that if $\alpha$ units enter $e$ then $\Gamma_e(\alpha)$ units leave it. We give a combinatorial algorithm for corresponding optimization problems, with running time polynomial in the network data and some simple parameters. We also exhibit new applications, showing that it is a general framework containing multiple convex programs for market equilibrum settings, for which combinatorial algorithms have been developed in the last decade. The result immediately settles some open problems, for example, giving a combinatorial algorithm for nonsymmetric Arrow-Debreu Nash bargaining proposed by Vazirani [38]. We can also extend existing results to more general settings.

Generalized flows are linear programs and thus can be solved efficiently by general linear programming techniques, the currently most efficient algorithm being the interior-point method by Kapoor and Vaidya [21]. Combinatorial approaches have been used since the sixties (e.g. [20, 25, 36]), yet the first

*College of Computing, Georgia Institute of Technology. Supported by NSF Grant CCF-0914732, and a Google Research Grant. E-mail: veghal@cs.elte.hu

polynomial-time combinatorial algorithms were given only in 1991 by Goldberg, Plotkin and Tardos [12]. This inspired a line of research to develop further polynomial-time combinatorial algorithms, e.g. [4, 14, 15, 35, 9, 16, 39, 30, 31]; for a survey on combinatorial generalized flow algorithms, see [32]. Our algorithm for this special case can be seen as a modified version of the algorithm by Goldfarb, Jin and Orlin [15].

As already revealed by early studies of the problem (e.g. [26, 36]), there is a deep connection between generalized flows and classical minimum-cost circulations: the dual structures are quite similar, and the generalized flow algorithms stem from the classical algorithms for minimum-cost circulations. We shall review these connections in Section 4. We shall also discuss a classical convex programming extension of minimum-cost circulations, when the cost $c_e$ of arc each arc is replaced by a convex function $C_e$. This problem has several applications and different algorithms for minimum-cost circulation were extended to this nonlinear setting, e.g. [23, 17, 22].

Nonlinear extensions of generalized flows have also been studied, e.g. in [1, 3], minimizing a separable convex cost function for generalized flows. However, these frameworks do not contain our problem, which involves nonlinear convex constraints.

We would like to argue that the relation between concave generalized flows and (linear) generalized flows is analogous to that between minimum-cost flows for separable convex and linear cost functions. That is, the convex setting is the most natural extension of the linear setting, with a potential extendability of the algorithms for the linear case.

Concave generalized flows being nonlinear convex programs, they can also be solved by the ellipsoid method, yet no efficient methods are known for this problem. Hence finding a combinatorial algorithm is also a matter of running time efficiency. Shigeno [33] gave the first combinatorial algorithm that runs in polynomial time for some restricted classes of functions $\Gamma_e$, including piecewise linear. It is an extension of the FAT-PATHS algorithm in [12]. In spite of this developement, it has remained an open problem to find a combinatorial polynomial-time algorithm for the general concave setting.

The starting point of our investigation is [15], also building on the approach of [17] for separable convex minimum-cost flows. The extension is natural, but some new insights are needed. A small, yet important first step is changing the perspective by considering a different, symmetric variant of the problem, that eliminates the distinguished role of the sink. Another important point is a new way of handling the cycle cancellation and excess transportation phases in a unified framework, done separately in most of the literature.

The concave optimization problem might have irrational optimal solutions: in general, we give a fully polynomial-time approximation scheme, with running time dependent on $\log(\frac{1}{\varepsilon})$ for finding an $\varepsilon$-approximate solution. In the market equilibrium applications we have rational convex programs (as in [38]): the existence of a rational optimal solution is guaranteed. We show a general technique to transform a sufficiently good approximation delivered by our algorithm to an exact optimal solution under certain circumstances. We demonstrate how this technique can be applied on the example of the linear Fisher market model.

In Section 2, we give the precise definition of the problems considered. Section 3 shows the applications for market equilibrium problems. Section 4 explores the background of minimum-cost circulation and generalized flow algorithms, and exhibits the main algorithmic ideas. Section 5 presents an algorithm for the symmetric version of the generalized flow problem. Based on this, Section 6 gives the algorithm for symmetric concave generalized flows. Section 7 adapts these algorithms for the usual sink formulation of the problems. Finally, Section 8 considers the case when the existence of a rational optimal solution is guaranteed, and shows how the approximate solution provided by our algorithm can be turned to an optimal solution.

## 2 Problem definitions

We define two closely related variants of the generalized flow problem and its concave extension. Let $G=(V,E)$ be a directed graph. Let $n=|V|$, $m=|E|$, and for each node $i\in V$, let $d_i$ be the total number of incoming and outgoing arcs incident to $i$.

We are given lower and upper arc capacities $\ell,u:E\to\mathbb{R}$ and gain factors $\gamma:E\to\mathbb{R}^+$ on the arcs, and node demands $b:V\to\mathbb{R}$. By a *pseudoflow* we mean a function $f:E\to\mathbb{R}$ with $\ell\le f\le u$. Given the pseudoflow $f$, let

$$e_i:=\sum_{j:ji\in E}\gamma_{ji}f_{ji}-\sum_{j:ij\in E}f_{ij}-b_i. \tag{1}$$

In the first variant of the problem, called the *sink formulation*, there is a distinguished node $t\in V$, called the *sink*. We assume $b_t=0$. The objective is to maximize $e_t$ for pseudoflows with $e_i\ge 0$ for all $i\in V-t$.

This differs from the way the problem is usually defined in the literature with the more restrictive $e_i=0$ for $i\in V-t$, and assuming $\ell\equiv 0$, $b\equiv 0$. However, this problem can easily be reduced to solving the sink formulation, see e.g. [32].

The following extension has been proposed by Truemper [37] and Shigeno [33]. On each arc $ij\in E$, we are given lower and upper arc capacities $\ell,u:E\to\mathbb{R}$ and a monotone increasing concave function $\Gamma_{ij}:\mathbb{R}\to\mathbb{R}\cup\{-\infty\}$; we are also given node demands $b:V\to\mathbb{R}$ with $b_t=0$. As for generalized flows, a pseudoflow is a function $f:E\to\mathbb{R}$ with $\ell\le f\le u$. For a pseudoflow $f$, let

$$e_i:=\sum_{j:ji\in E}\Gamma_{ji}(f_{ji})-\sum_{j:ij\in E}f_{ij}-b_i.$$

In the *concave sink formulation*, the objective is to maximize $e_t$ for pseudoflows so that so that $e_i\ge 0$ for all $i\in V-t$.

Shigeno [33] defines this problem with $e_i=0$ if $i\in V-t$, and $b\equiv 0$ and without explicit capacity constraints, but also discusses the version with $e_i\ge 0$. She also gives a reduction from the original version to this one. Whereas capacity constraints can be simulated by the functions $\Gamma_e$, we impose them explicitly as they will be included in the running time estimations. The formulation with $e_i\ge 0$ seems more natural as it gives a convex optimization problem, which is not the case for $e_i=0$.

In the sink formulation, the node $t$ plays a distinguished role. It turns out to be more convenient for our algorithmic purposes to handle all nodes equally. For this reason, we introduce another, seemingly more general version, called the *symmetric formulation* of both problems. Ideally, we would like to find a pseudoflow satisfying $e_i\ge 0$ for every $i\in V$. The formulation will be a relaxation of this feasibility problem, allowing violation of the constraints, penalized by possibly different rates at different nodes.

For each node $i\in V$ we are given a penalty factor $M_i>0$ and an auxiliary variable $\kappa_i\ge 0$. The objective is to minimize $\kappa_f=\sum_{i\in V}M_i\kappa_i$ for a pseudoflow $f$ subject to $e_i+\kappa_i\ge 0$ for each $i\in V$.

The objective $\kappa_f$ is called the *excess discrepancy*. $\kappa_f=0$ would mean $e_i\ge 0$ for each $i\in V$. These conditions might be violated, but we have to pay penalty $M_i$ per unit for violation at $i$.

The sink version fits into this framework with $M_i=\infty$ for $i\ne t$ and $M_t=1$. Yet for algorithmic reasons, we will allow only finite $M_i$ values. Still we can approximate the sink version by setting high $M_i$ values for $i\ne t$ and $M_t=1$. We shall show that for generalized flows, solving the symmetric version with high enough $M_i$ values delivers an optimal solution for the sink version. Besides the sink version, another natural setting is when $M_i=1$ for all $i\in V$, that is, maintaining $e_i\ge 0$ has the same importance for all nodes.

While the symmetric formulation could seem more general as the sink version, it can indeed be reduced to it. For an instance of the symmetric version with graph $G = (V, E)$, let us add a new node $t$ with an arc from $t$ to every node $i \in V$ with gain factor $1/M_i$. Solving the sink version for this extended instance gives an optimal solution to the original problem. The reason for introducing the symmetric formulation is that it is more suitable for our algorithmic purposes.

### 2.1 Complexity assumptions

The complexity setting will be different for generalized flows and concave generalized flows. For generalized flows, we aim to find an optimal solution, while in the convave case, only an approximate one. For generalized flows, the gain functions are given explicitly as linear functions, while in the concave case, the description of the functions might be infinite. To handle this difficulty, following the approach of Hochbaum and Shantikumar [17], we assume oracle access to the $\Gamma_{ij}$'s: our running time estimation will give a bound on the number of necessary oracle calls. Two kinds of oracles are needed: (i) value oracle, returning $\Gamma_{ij}(\alpha)$ for any $\alpha \in [\ell_{ij}, u_{ij}]$; and (ii) inverse value oracle, returning a value $\beta$ with $\alpha = \Gamma_{ij}(\beta)$ for any $\alpha \in [\Gamma_{ij}(\ell_{ij}), \Gamma_{ij}(u_{ij})]$.

We assume that both oracles return the exact (possibly irrational) solution, and any oracle query is done in $O(1)$ time. Also, we assume any basic arithmetic operation is performed in $O(1)$ time, regardless to size and representation of the possibly irrational numbers. We expect that our result would naturally extend to setting with only approximate oracles and computational capacities. Notice that in an approximate sense, an inverse value oracle can be simulated by a value oracle.

By an $\varepsilon$-approximate solution to the symmetric concave generalized flow problem we mean a feasible solution with the excess dicrepancy larger than the optimum by at most $\varepsilon$. An $\varepsilon$-approximate solution to the sink version means a solution with the objective value $e_t$ at most $\varepsilon$ less than the optimum, and the total violation of the inequalities $e_i \geq 0$ for $i \in V - t$ is also at most $\varepsilon$.

In both cases, we assume that all $M_i$ values are positive integers, and let $M$ denote their maximum.

For generalized flows, we assume all data $(\ell, u, b, \gamma)$ is given as rational numbers; let $B$ be the largest integer used in the description. The running time bound will be $O(m^2(m \log B + \log M) \log m)$ for the symmetric formulation and $O(m^3 \log m \log(d_t B))$ for the sink formulation. This is slightly worse than the bound of [15]; it could be improved if we would not transform the problem to an uncapacitated instance. Yet we perform this transformation in order to simplify the presentation.

For the concave setting, we allow irrational capacities as well; in the complexity estimation, we will have $U$ as an upper bound on the absolute values on the $b_i$'s, the capacities $\ell_{ij}, u_{ij}$ and the $\Gamma_{ij}(\ell_{ij})$, $\Gamma_{ij}(u_{ij})$ values. For each arc $ij$, let us define $r_{ij} = |\Gamma_{ij}(\ell_{ij})|$ whenever $\Gamma_{ij}(\ell_{ij}) > -\infty$ and $r_{ij} = 0$ otherwise. Let

$$U = \max\left\{\max\{|b_i| : i \in V\}, \max\{|\ell_{ij}|, |u_{ij}|, |\Gamma_{ij}(u_{ij})|, r_{ij} : ij \in E\}\right\}.$$

For the symmetric version, our running time bound will be $O(m(m + n \log n) \log(MUm/\varepsilon))$.

For the sink version, we need to introduce one further complexity parameter $U^*$ due to difficulties arising if $\Gamma_{ij}(\ell_{ij}) = -\infty$ for certain arcs. Let $U^*$ satisfy $U \leq U^*$, and that $e_t \leq U^*$ for any pseudoflow (it is easy to see that $U^* = d_t U$ always satisfies this property). We also require that whenever there exists a feasible solution to the problem (that is, $e_i \geq 0$ for each $i \in V - t$), there exists one with $e_t \geq -U^*$.

If $\Gamma_{jt}(\ell_{jt}) > -\infty$ for each arc $jt \in E$, then $U^* = d_t U$ also satisfies this property. However, for the linear Fisher market model, all arcs $it$ have $\Gamma_{it}(\ell_{it}) = -\infty$; this will be discussed in Section 8.

For the sink version, the running time bound will be $O(m(m + n \log n) \log(U^* m/\varepsilon))$.

## 3 Applications to market equilibrium problems

Intensive research has been pursued in the last decade to develop polynomial-time combinatorial algorithms for certain market equilibrium problems. The starting point is the algorithm for computing market clearing prices in Fisher's model with linear utilities by Devanur, Papadimitriou, Saberi, Vazirani [6], followed by a study of several variations and extensions of this model. For a survey, see [24, Chapter 5] or [38].

The equilibrium solution for linear Fisher markets was described via a convex program by Eisenberg and Gale [8] in 1959; the combinatorial algorithms for this problem and other models rely on the KKT-conditions for the corresponding convex programs. Exact optimal solutions can be found, since these problems admit rational optimum solutions.

We show that the Eisenberg-Gale convex program, along with all extensions studied so far, falls into the broader class of convex generalized flows. Moreover, in all these extension we may replace linear or piecewise linear concave functions by arbitrary concave functions, to obtain general settings, still solvable approximately by our algorithm.

In the *linear Fisher market model*, we are given a set $B$ of buyers and a set $G$ of goods. Buyer $i$ has a budget $m_i$, and there is one divisible unit of each good to be sold. For each buyer $i \in B$ and good $j \in G$, $U_{ij} \geq 0$ is the utility accrued by buyer $i$ for one unit of good $j$. Let $n = |B| + |G|$ and $m$ be the number of pairs $ij$ with $U_{ij} > 0$.

An equilibrium solution consist of prices $p_i$ on the goods and an allocation $x_{ij}$, so that (i) all goods are sold, (ii) all money of the buyers is spent, and (iii) each buyers $i$ buys a best bundle of goods, that is, goods $j$ maximizing $u_{ij}/p_j$. The optimal solution to the *Eisenberg-Gale convex program* gives such an allocation:

$$\begin{aligned}
\max \sum_{i \in B} m_i \log z_i & \\
z_i \leq \sum_{j \in G} U_{ij} x_{ij} & \qquad \forall i \in B \\
\sum_{i \in B} x_{ij} \leq 1 & \qquad \forall j \in G \\
x_{ij} \geq 0 & \qquad \forall i \in B, j \in G
\end{aligned}$$

Let us define the graph $(V, E)$ with $V = B \cup G \cup \{t\}$ by adding a sink node $t$. Add an arc $ji$ whenever $j \in G$, $i \in B$, $U_{ij} > 0$, and set $\Gamma_{ji}(\alpha) = U_{ij}\alpha$ as a linear gain function. Add an arc $it$ for any $i \in B$ with $\Gamma_{it}(\alpha) = m_i \log \alpha$. Finally, set $b_j = -1$ for $j \in G$, and $b_i = 0$ for $i \in B$. The above program describes exactly the sink version of this concave generalized flow instance with $f_{ji} = x_{ij}$ for $i \in B$, $j \in G$ and $f_{it} = z_i$. (To formally fit into the model, we may add upper capacities $u_{ji} = 1$ and $u_{it} = \sum_{j \in G} U_{ji}$ without changing the solution.) Hence our general algorithm gives a $\varepsilon$-approximation for this problem. In Section 8, we show that for sufficiently small $\varepsilon$ we can transform it to an optimal solution, providing an $O(mn(m+n\log n)\log U_{\max})$ algorithm. The most efficient currently known algorithm is $O(n^4 \log n)$ by Orlin [27].

The flexibility of the concave generalized flow model enables us various extensions. For example, we can replace each linear function $U_{ji}\alpha$ by an arbitrary concave increasing function, yielding the perfect price discrimination model of Goel and Vazirani [11]. They studied piecewise linear utility functions; our model enables arbitrary functions (although a rational optimal solution does not necessarily exist anymore).

In the *Arrow-Debreu Nash bargaining (ADNB)* game defined by Vazirani [38], there is a disagreement point, with a utility $c_i \geq 0$ guaranteed for player $i$. In an equilibrium solution, we price the goods and distribute them among the buyers so (i) all goods are sold; (ii) each buyer $i$ spends her entire budget $m_i = 1$; (iii) every buyer buys an optimal bundle of goods; and (iv) every buyer accrues a utility greater than $c_i$.

Unlike for the linear Fisher model, equilibrium prices may not exist, corresponding to a disagreement solution. [38] gives a sophisticated two phase algorithm, first for deciding feasibility, then finding the equilibrium solution. The algorithm heavily relies on the fact that the budget of every buyer is the same. Finding a combinatorial algorithm for the case with arbitrary budgets $m_i$, called the *nonsymmetric ADNB* game, was left open.

The convex program for nonsymmetric ADNB can be obtained from the Eisenberg-Gale program by modifying the first set of inequalities to $z_i \leq \sum_{j \in G} U_{ij} x_{ij} - c_i$. In the formulation as a concave generalized flow, this corresponds to modifying the $b_i = 0$ values for $i \in B$ to $b_i = c_i$. Hence this problem also fits into our framework; moreover, the nonsymmetric case is not more difficult than the symmetric case. From this general perspective, it does not even seem to be harder than the linear Fisher model.

Another open question in [38] is to devise a combinatorial algorithm for (nonsymmetric) ADNB with piecewise linear, concave utility functions. Our result generalizes even further, for arbitrary concave utility functions, since the linear functions $U_{ij}\alpha$ can be replaced by arbitrary concave functions.

Let us also remark that an alternative convex program for the linear Fisher market, given by Shmyrev [34], shows that it also fits into the framework of minimum-cost circulations with separable convex cost function, and thus can be solved by the algorithms of Hochbaum and Shantikumar [17] or Karzanov and McCormick [22]. However, this does not seem to be the case for ADNB, where no alternative formulation analogous to [34] is known.

As further applications of the concave generalized flow model, we can take single-source multiple-sink markets by Jain and Vazirani [19], or concave cost matchings studied by Jain [18].

A distinct characteristic of the Eisenberg-Gale program and its extensions is that they are rational convex programs. This property can get lost when changing to general concave spending constraint utilities. Yet for the case when the existence of a rational solution is guaranteed, one would prefer finding an exact optimal solution. Section 8 addresses the question of rationality. Theorem 8.1 shows that under certain technical conditions, our approximation algorithm can be turned into a polynomial time algorithm for finding an exact optimal solution. The conditions are natural: we shall verify them only for the linear Fisher market, but they should also hold for all aforementioned rational convex programs.

# 4 Background

Minimum-cost circulations are fundamental to all problems and algorithms discussed in the paper. We give an overview on them in Section 4.1. We present the two main algorithmic paradigms, cycle cancelling and successive shorthest paths along with their efficient variants. In Section 4.2, we continue with an overview of generalized flow algorithms, exhibiting some important ideas and their relation to minimum-cost flows. Section 4.3 considers a different convex extension of minimum-cost circulations, when the linear cost function is replaced by a separable convex one. We show how the two main paradigms extend to this case, using different approximation strategies of the nonlinear functions. Finally in Section 4.4 we consider the concave generalized flow problem, discuss the algorithm by

Shigeno [33] and its relation to algorithmic ideas of the previous problems. We emphasize some difficulties and sketch the ideas for solving them.

### 4.1 Minimum-cost flows: cycle cancelling and sucessive shortest paths

In this section, we give a brief and simplified overview of minimum-cost circulation algorithms. Given is a directed graph $G = (V, E)$ with lower and upper arc capacities $\ell, u : E \to \mathbb{R} \cup \{\infty\}$, costs $c : E \to \mathbb{R}$ on the arcs and node demands $b : V \to \mathbb{R}$ with $\sum_{i \in V} b_i = 0$. Let

$$e_i = \sum_{j:ji \in E} f_{ji} - \sum_{j:ij \in E} f_{ij} - b_i.$$

$f : E \to \mathbb{R}$ with $\ell \le f \le u$ is called a feasible circulation, if $e_i = 0$ for all $i \in V$. The objective is to minimize $c^T f$ for feasible circulations.

Linear programming duality theory provides the following characterization of optimality. For a feasible circulation $f$, let us define the residual graph $G_f = (V, E_f)$ with $ij \in E_f$ if $ij \in E$ and $f_{ij} < u_{ij}$, or if $ji \in E$ and $\ell_{ji} < f_{ji}$. The first type of arcs are called forward arcs and are assigned the original cost $c_{ij}$, while the latter arcs are backward arcs assinged cost $-c_{ji}$. For notational convenience, we will use $f_{ij} = -f_{ji}$ on backward arcs. Then $f$ is optimal if and only if $E_f$ contains no negative cost cycles. This is further equivalent to the existence of a feasible potential $\pi : V \to \mathbb{R}$ with $\pi_j - \pi_i \le c_{ij}$ for all arcs $ij \in E_f$.

Two main frameworks for minimum-cost flow algorithms are as follows. In the *cycle cancelling framework* (see e.g. [2, Chapter 9.6]), we maintain a feasible circulation in each phase, with strictly increasing objective values. If the current solution is not optimal, the above conditions guarantee a negative cost cycle in the residual graph. A negative cost cycle can be found efficiently. Sending some flow around this cycle decreases the objective and maintains feasibility, providing the next solution.

In the *sucessive shortest path framework* (see e.g. [2, Chapter 9.7]), we waive feasibility by allowing $e_i < 0$ and $e_i > 0$; we call such nodes positive and negative, respectively. However, we maintain dual optimality in the sense that the residual graph of the current pseudoflow contains no negative cost cycles in any iteration (or equivalently, admits a feasible potential). If there exists some positive and negative nodes, we send some flow from a positive node to a negative one using a minimum-cost path in the residual graph. This maintains dual optimality, and decreases the total $e_i$ value of positive nodes.

For rational input data, both these algorithms are finite, but may take an exponential number of steps (and might not even terminate for irrational input data). Nevertheless, using (explicit or implicit) scaling techniques, both can be implemented to run in polynomial time, and even in strongly polynomial time.

A strongly polynomial version of the cycle cancellation algorithm is due to Goldberg and Tarjan [13]. In each step, a minimum mean cycle is chosen. In dual terms, we relax primal-dual optimality conditions to $\pi_j - \pi_i \le c_{ij} + \varepsilon$ for all $ij \in E_f$, with $\varepsilon$ being equal to the negative of the minimum mean cycle value, decreasing exponentially over time.

Polynomial implementations of the sucessive shortest path algorithm can be obtained by capacity scaling; the most efficient, strongly polynomial such algorithm is due to Orlin [28]. We describe here a basic capacity scaling framework by Edmonds and Karp [7]. Instead of the residual graph $E_f$, we consider the $\Delta$-residual graph $E_{\Delta,f}$ consisting of arcs with residual capacity at least $\Delta$ (the residual capacity is $u_{ij} - f_{ij}$ on a forward arc $ij$ and $f_{ji} - \ell_{ji}$ on a backward arc). The algorithm consists of $\Delta$-scaling phases, with $\Delta$ decreasing exponentially. In a $\Delta$-phase, we iteratively send $\Delta$ units of flow from a positive node $s$ with $e_s \ge \Delta$ to a negative node $t$ with $e_t \le -\Delta$ on a minimum-cost path of

$E_{\Delta,f}$. The $\Delta$-phase finishes when this is no longer possible, which means the total positive excess is at most $n\Delta$.

In the $\Delta$-phase, $\pi_j - \pi_i \leq c_{ij}$ is maintained on arcs of the $\Delta$-residual graph. When moving to the $\Delta/2$ phase, this might not hold anymore, since the $\Delta/2$-residual graph contains more arcs, those with residual capacity between $\Delta/2$ and $\Delta$. At the beginning of the next phase, we saturate all these arcs, thereby increasing the positive excess to at most $(2n+m)\Delta/2$. This guarantees that the next phase will consist of at most $(2n+m)$ path augmentations.

### 4.2 Generalized flows – cycle canceling and excess transportation

In what follows, we consider the sink version of the generalized flow problem, with sink $t \in V$. For a pseudoflow $f : E \to \mathbb{R}$, let us define the residual network $G_f = (V, E_f)$ as for circulations, with gain factor $\gamma_{ij} = 1/\gamma_{ji}$ on backward arcs. Consider a cycle $C = \{i_0, i_1, \ldots, i_{k-1}\}$ in $E_f$. We can modify $f$ by sending some flow around $C$ from $i_0 \in C$ in the following sense: we start sending $\alpha > 0$ units on $i_0 i_1$; the flow arriving at $i_1$ is $\gamma_{i_0 i_1}\alpha$. On each subsequent arc we send the amount that arrived from the previous arc. This leaves $e_j$ unchanged if $j \neq i_0$, and increases $e_{i_0}$ by $\gamma(C)\alpha$, where $\gamma(C) = \Pi_{e\in C}\gamma_e$. If $\gamma(C) > 1$ then we call $C$ a *flow-generating cycle*, while for $\gamma(C) < 1$, a *flow-absorbing cycle*, since we can generate or eliminate excess at an arbitrary node $i_0 \in C$, respectively. The amount of flow that can be generated is of course bounded by the capacity constraints.

To augment the excess of the sink, we have to send the excess generated at a flow-generating cycle $C$ to the sink $t$. Hence we call a pair $(C, P)$ a *generalized augmenting path (GAP)*, if (a) $C$ is a flow-generating cycle, $i_0 \in V(C)$, and $P$ is a path in $E_f$ from $i_0$ to $t$; or (b) $C = \emptyset$, and $P$ is a path in $E_f$ from some node $i_0$ with $e_{i_0} > 0$ to $t$. Clearly, an optimal solution $f$ may admit no GAPs. This is indeed equivalence: $f$ is optimal if and only if no GAP exists.

The gain factors $\gamma_e$ play a role analogous to the costs $c_e$ for minimum-cost circulations. Indeed, $C$ is a flow generating cycle if and only if it is a negative cost cycle for the cost function $c_e = -\log\gamma_e$. The dual structure for generalized flows is also analogous to potentials. Let us call $\mu : V \to \mathbb{R}_{>0} \cup \{\infty\}$ with with $\mu_t = 1$ a label function. *Relabeling* the pseudoflow $f$ by $\mu$ means multiplying the flow on each arc $ij$ going out from $i$ by $\mu_i$. We get a problem equivalent to the original by replacing each arc gain by $\gamma^{\mu}_{ij} = \gamma_{ij}\mu_i/\mu_j$. The labeling is called *conservative* if $\gamma^{\mu}_{ij} \leq 1$ for all $ij \in E_f$, that is, no arc may increase the relabeled flow.

Assume we have a conservative labeling $\mu$ so that $e_i = 0$ whenever $i \in V - t$, $\mu_i < \infty$. Let $V' \subseteq V$ denote the set of nodes from which there exists a directed path to $t$. It follows that (i) $\mu_i < \infty$ for all $i \in V'$, and (ii) $V'$ contains no flow-generating cycles. Consequently, given a conservative labeling, no GAP can exist, and the converse can also be shown to hold. Note that on $V'$, $\pi_i = -\log\mu_i$ is a feasible potential for $c_e = -\log\gamma_e$ with $\pi_t = 0$ if and only if $\mu$ is conservative.

Based on this correspondence, minimum-cost circulation algorithms can be directly applied for generalized flows as a subroutine for eliminating all flow-generating cycles. This can be indeed implemented in strongly polynomial time, see [29, 32]. The novel difficulty for generalized flows is how to transport the generated excess from various nodes of the graph to the sink $t$. In the algorithm of Onaga [26], flow is transported iteratively on highest gain augmenting paths, that is, from $i \in V$ with $e_i > 0$ on an $i - t$ path $P$ that maximizes $\gamma(P) = \Pi_{e\in P}\gamma_e$. It can be shown that using such paths does not create any new flow generating cycles. Thus after having eliminated all type (a) GAPs, we only have to take care of type (b). Unfortunately, this algorithm may run in exponential time (or may not even terminate for irrational inputs). This is due to the analogy between Onaga's algorithm and the sucessive shortest path algorithm – observe that a highest gain path is a minimum-cost path for

$-\log \gamma_e$.

The first algorithms to overcome this difficulty and thus establish polynomial running time bounds were the two given by Goldberg, Plotkin and Tardos [12]. One of them, the FAT-PATHS algorithm, uses a method analogous to capacity scaling. A path $P$ in $E_f$ from a node $i$ to $t$ is called $\Delta$-fat, if assuming unlimited excess at $i$, it is possible to send enough flow along $P$ from $i$ to $t$ so that $e_t$ increases by $\Delta$.

The algorithm consists of $\Delta$-phases, with $\Delta$ decreasing by a factor of 2 for the next phase. In the $\Delta$-phase, we first cancel all flow generating cycles. Then, from nodes $i$ with $e_i > 0$, we transport flow on highest gain ones among the $\Delta$-fat paths. This might create new flow-generating cycles to be cancelled in the next phase. Nevertheless, it can be shown that at the beginning of a $\Delta$-phase, $e_t^* - e_t \leq 2(n+m)\Delta$ for the optimum value $e_t^*$ and thus the number of path augmentations in each $\Delta$-phase can be bounded by $2(n+m)$. Arriving at a sufficiently small value of $\Delta$, it is possible to obtain an optimal solution by a single maximum flow computation.

The basic framework of [26] and of FAT-PATHS, namely using different subroutines for eliminating flow-generating cycles and for tranporting excess to the sink has been adopted by most subsequent algorithms, e.g. [14, 15, 35, 9, 30].

The algorithm that serves as a starting point of our approach is by Goldfarb, Jin and Orlin [15]. As in Onaga's algorithm [26], a single cycle canceling subroutine at the beginning suffices: augmentation indeed happens always on highest gain paths, and thus no new flow-generating cycles appear in the later course of the algorithm. The algorithm consists of $\Delta$-phases. In a $\Delta$-phase, we try to send $\Delta$ units of flow from some node with positive excess to the sink on a highest gain path $P$. The residual capacity of some arcs of $P$ might be insufficient for this augmentation. If the residual capacity of arc $ij \in E_f$ is too small to carry the necessary flow, we send only as much as necessary to saturate $ij$, and leave the rest in node $i$, in a special repository assigned to $ij$, called an arc imbalance. (Arc imbalances were introduced in [14]).

At the end of the $\Delta$-phase, arc imbalances are converted to node excesses. The key observation is that the total arc imbalances accrued in the entire $\Delta$-phase is less than $\Delta$, and thus the total excess at the beginning of the next scaling phase is at most $(n+2m)\Delta$.

In our approach, the main difference is that cycle cancellation and excess transportation are handled in a unified framework. In the capacity scaling algorithm for minimum-cost circulations, cycle cancelation is done by path augmentations: analogously, our algorithm only uses path augmentations which eliminate flow-generating cycles and transport excess simultaneously.

In the sucessive shortest paths framework, we start with an infeasible pseudoflow, having positive and negative nodes. To use an analogous method for generalized flows, we have to give up the standard framework of algorithms where $e_i \geq 0$ is always maintained for all $i \in V - t$. This is the reason why we use the more flexible symmetric model: we start with possibly several nodes having $e_i < 0$, and our aim is to eliminate them. An important property of the algorithm is that we always have to maintain $\mu_i = 1/M_i$ for $e_i < 0$; for this reason we shall avoid creating new negative nodes.

Our generalized flow algorithm, described in Section 5, consists of $\Delta$-phases. In a phase, we transport $\Delta$ units of relabeled flow from a node with relabeled excess at least $\Delta$ to a node with negative excess on a highest gain path. We also use arc imbalances, in a simplified manner enabled by initially transforming the problem to an equivalent one with no upper capacities on arcs. For a sufficiently small value of $\Delta$ we can obtain an optimal solution by a single maximum flow computation, a standard method in the generalized flow literature.

For the sink version, described in Section 7, we perform this algorithm with $M_i = B^n + 1$ if $i \neq t$ and $M_t = 1$. We shall show that this returns an optimal solution.

### 4.3 Minimum-cost circulations with separable convex costs

A natural and well-studied nonlinear extension of minimum-cost circulations is replacing each arc cost $c_e$ by a convex function $C_e$. We are given a directed graph $G = (V, E)$ with lower and upper arc capacities $\ell, u : E \to \mathbb{R}$, convex cost functions $C_e : [\ell_e, u_e] \to \mathbb{R}$ on the arcs, and node demands $b : V \to \mathbb{R}$ with $\sum_{i \in V} b_i = 0$. Our aim is to minimize $\sum_{e \in E} C_e(f_e)$ for feasible circulations $f$. This is a widely applicable framework, see [2, Chapter 14].

This is a convex optimization problem, and optimality can be described by the KKT conditions. Let $C_e^-(\alpha)$ and $C_e^+(\alpha)$ denote the left and right derivatives of $C_e$ in $\alpha$, respectively. As before, for a feasible circulation $f$ define the auxiliary graph $G_f = (V, E_f)$. Let $C_{ij}$ denote the original function if $ij$ is a forward arc and let $C_{ji}(\alpha) = -C_{ij}(-\alpha)$ on backward arcs. Notice that $C_{ji}^-(f_{ji}) = -C_{ij}^+(f_{ij})$. $f$ is optimal if and only if there exists no cycle $C$ in $E_f$ with $\sum_{e \in C} C_e^+(f_e) < 0$. In dual terms, $f$ is optimal if and only if there exists a potential $\pi : V \to \mathbb{R}$ with $\pi_j - \pi_i \le C_{ij}^+(f_{ij})$ for all $ij \in E_f$.

Both the minimum mean cycle cancelation and the capacity scaling algorithms can be naturally extended to this problem with polynomial (but not strongly polynomial!) running time bounds. However, these two approaches relax the optimality conditions in fundamentally different ways.

Cycle cancelation was adapted by Karzanov and McCormick [22]. The algorithm subsequently cancels cycles in $E_f$ with minimum mean value respect to the $C_e^+(f_e)$ values. The only difference is that the flow augmentation around such a cycle might be less than what residual capacities would enable, in order to maintain

$$\pi_j - \pi_i \le C_{ij}^+(f_{ij}) + \varepsilon \ \forall ij \in E_f \tag{2}$$

for the current potential $\pi$ and scaling parameter $\varepsilon$.

For capacity scaling, Hochbaum and Shanthikumar [17] developed the following framework based on previous work of Minoux [23] (see also [2, Chapter 14.5]). The algorithm consists of $\Delta$-phases. In the $\Delta$-phase, each $C_e$ is linearized with granularity $\Delta$.

Let $E_{\Delta,f}$ denote the $\Delta$-residual network. We will maintain $\Delta$-optimality, that is, there exists a potential $\pi$ such that

$$\pi_j - \pi_i \le \frac{C_{ij}(f_{ij} + \Delta) - C_{ij}(f_{ij})}{\Delta} \ \forall ij \in E_{\Delta,f}. \tag{3}$$

Let $\theta_\Delta(ij)$ denote the quantity on the right hand side. If we increase flow on some arc $ij$ by $\Delta$ on some $ij$ for which equality holds, the resulting pseudoflow will remain $\Delta$-optimal. We will always send $\Delta$ units of flows from a node $s$ with $e_s > 0$ to a node $t$ with $e_s < 0$ on a minimum-cost path in $E_{\Delta,f}$ with respect to $\theta_\Delta(ij)$. By the above observation, this maintains $\Delta$-optimality.

When moving to the next scaling phase replacing $\Delta$ by $\Delta/2$, we change to a better linear approximation of the $C_e$'s. Therefore, (3) may get violated not only because $E_{\Delta/2,f}$ contains more arcs than $E_{\Delta,f}$, but also on arcs already included in $E_{\Delta,f}$. Yet it turns out that modifying each $f_{ij}$ value by at most $\Delta/2$, (3) can be re-established. This creates new (positive and negative) excesses of total at most $m\Delta$.

### 4.4 Concave generalized flows

As we have seen, both the cycle canceling and capacity scaling approaches for minimum-cost circulations naturally extend to separable concave cost functions. Similarly, our algorithm in Section 6 for concave gain functions can be seen as a natural extension of the generalized flow algorithm in Section 5.

However, this extension is by no means obvious. We were not able to extend any of the previously existing generalized flow algorithms: although our version is very similar to that of Goldfarb, Jin and

Orlin [15], there is the important difference of eliminating the initial cycle canceling phase. Also, the extension to the concave case needs the new idea of reserving some of the node excess for later adjustments on incident arcs.

Shigeno's [33] approach was to extend the Fat-Paths algorithm of [12]. However, she could obtain polynomial running time bounds only for restricted classes of gain functions. The algorithm consists of two procedures applied alternately similar to Fat-Paths: a cycle cancelation phase to generate excess on cycles with positive gains, and a path augmentation phase to transport new excess to the sink in chunks of $\Delta$. For both phases, previous methods naturally extend: cycle canceling is performed analogously to [22], whereas path augmentation to [17]. Unfortunately, fitting the two different methods together is problematic and does not yield polynomial running time.

The main reason is that the two approaches rely on fundamentally different kinds of approximation of the nonlinear gain functions. While for generalized flows, a cycle canceling phase completely eliminates flow generating cycles, here we can only get an approximate solution allowing some small positive gain on cycles. In terms of residual arcs, we terminate with a condition analogous to (2) for concave cost flows. However, the path augmentation phase needs a linearization of the gain functions analogous to (3). Notice that for $\varepsilon = 0$, (2) implies (3) for arbitrary $\Delta$. Yet if some small error $\varepsilon > 0$ is allowed, then no general guarantee can be given so that (3) hold for a certain value $\Delta$.

The crucial difficulty in the approach of [33] is using these two different kind of approximations of the dual optimality condition simultaneously. It is not obvious how to avoid this, since for generalized flows, all previous approaches for generating excess on cycles use a cycle cancelation approach, while all methods for transporting the excess to the sink use a scaling approach. Hence implicitly or explicitly, all such two-phase algorithms use the two different paradigms.

Since no natural method in the cycle cancelation framework seems to work for transporting excess, a possible way out is to try to extend the scaling method for the entire algorithm, not using any approximation of derivatives as in (2). Yet, scaling methods for minimum-cost flows presume several sinks and sources in the graph, while for generalized flows, only one sink is allowed. As discussed in Section 4.2, we resolve this difficulty by changing the problem setting from the sink version to the symmetric version, eliminating the distinguished role of the sink. Using this idea, we can completely eliminate the (single) cycle canceling phase from [15] and integrate excess generation into the framework of paths augmentations.

For concave gains there is a further, serious issue with linearization analogous to (3). When moving from a $\Delta$-phase to a $\Delta/2$-phase, we change to a better linear approximation of the gain functions, and thus some arcs may become infeasible. Feasibility can be restored by changing the flow on each arc by a small amount, modifying the node excesses. This is no problem for circulations since the total modification is limited. However, for generalized flows, we must avoid creating new nodes with negative excess, since for dual optimality we have to maintain $\mu_i = 1/M_i$ whenever $e_i < 0$, a condition possibly violated by creating new negative nodes.

In the concluding section of [33], Shigeno already proposed to try adapting the method of [15] for concave generalized flows, in order to avoid repeatedly calling the cycle cancelation procedure, and pointed out the above difficulty. We resolve this by maintaing a security reserve of $d_i\Delta$ in each node $i$. This gives an upper bound on the total change caused by restoring feasibility of incident arcs in all subsequent phases. We maintain the stronger condition $\mu_i = 1/M_i$ for all $e_i < d_i\Delta$, that is, all nodes who can possibly become negative at any later point. For this reason, we might treat some nodes with positive excesses as sinks and send flow to them. Later as $\Delta$ decreases, such nodes gradually become sources.

## 5 Generalized flows algorithm

In this section, we investigate the symmetric formulation of the generalized flow problem. In describing the optimality conditions, we also allow infinite $M_i$ values to encorporate the sink version. However, in the algorithmic parts, we restrict ourselves to finite $M_i$ values.

We describe optimality conditions in Section 5.1. Notion and results here are well-known in the generalized flow literature, thus we do not include references. Section 5.2 describes the subroutine TIGHTEN-LABEL, which is essentialy Dijkstra's algorithm for computing highest-gain paths. The main algorithm is exhibited in Section 5.3.

The algorithm works on an uncapacitated graph, since this setting allows certain simplifications both in the presentation and in the analysis. The description of the transformation is postponed after the algorithm to Section 5.4. The final step of the algorithm is presented in Section 5.5, where we show that when the total relabeled excess is sufficiently small, an optimal solution can be found by a single maximum flow computation. Analysis and running time bounds are given in Section 5.6.

### 5.1 Optimality conditions

Given the network $G = (V, E)$ with lower and upper capacities $u, \ell$ and a pseudoflow $f$, we define the residual network $G_f = (V, E_f)$ as follows. Let $ij \in E_f$, if $ij \in E$ and $f_{ij} < u_{ij}$ or if $ji \in E$ and $f_{ji} > \ell_{ji}$. The first type of arcs are called *forward arcs*, while the second type are the *backward arcs*. In the algorithm, we will use a network without upper capacities; in this case, every $ij \in E$ is included in $E_f$. For a forward arc $ij$, let $\gamma_{ij}$ be the same as in the original graph. For a backward arc $ji$, let $\gamma_{ji} = 1/\gamma_{ij}$. Also, we define $f_{ji} = -\gamma_{ij} f_{ij}$ for every backward arc $ji \in E_f$. By increasing (decreasing) $f_{ji}$ by $\alpha$, we mean decreasing (increasing) $f_{ij}$ by $\alpha/\gamma_{ij}$.

Let $P = i_0 \dots i_k$ be a walk in the auxiliary graph $E_f$. By sending $\alpha$ units of flow along $P$, we mean increasing each $f_{i_h i_{h+1}}$ by $\alpha \Pi_{0 \le t < h} \gamma_{i_t i_{t+1}}$. We assume $\alpha$ is chosen small enough so that no capacity gets violated. Let $f^{\alpha,P}$ denote the modified flow. Note that this decreases $e_{i_0}$ by $\alpha$, increases $e_{i_k}$ by $\alpha \Pi_{0 \le t < k} \gamma_{i_t i_{t+1}}$, and leaves the other $e_i$ values unchanged.

Let $C = i_0 \dots i_{k-1}$ be a cycle in $E_f$. Then by sending $\alpha$ units of flow around $C$ from $i_0$ we mean sending $\alpha$ units on the walk $i_0 \dots i_{k-1} i_0$. This modifies only $e_{i_0}$, increasing by $(\gamma(C) - 1)\alpha$ for $\gamma(C) = \Pi_{e \in C} \gamma_e$. $C$ is called a *flow generating cycle* if $\gamma(C) > 1$. On such a cycle for any choice of $i_0 \in V(C)$, we can create an excess of $(\gamma(C) - 1)\alpha$ by sending $\alpha$ units around (assuming that $\alpha$ is sufficiently small so that no capacity constraints are violated).

The pair $(C, P)$ is called a *generalized augmenting path (GAP)* in the following cases:

(a) $C$ is a flow generating cycle, $i_0 \in V(C)$, $t \in V$ is a node with $e_t < 0$, and $P$ is a path in $E_f$ from $i_0$ to $t$ ($i_0 = t$, $P = \emptyset$ is possible);

(b) $C = \emptyset$, and $P$ is a path between two nodes $s$ and $t$ with $e_s > 0$, $e_t < 0$;

(c) $C = \emptyset$, and $P$ is a path between $s$ and $t$ with $e_s \le 0$, $e_t < 0$ and $\gamma(P) = \Pi_{e \in C} \gamma_e > M_s/M_t$.

**Lemma 5.1.** *If $f$ is an optimal solution, then no GAP exists.*

*Proof.* In case (a), we can send some $\alpha > 0$ units of flow around $C$ from $i_0$, and then send the generated $(\gamma(C)-1)\alpha$ excess from $i_0$ to $t$ along $P$. For sufficiently small positive value of $\alpha$, this is possible without violating the capacity constraints and it decreases the excess discrepancy. In case (b), we can decrease the excess discrepancy at $t$ while only decreasing a positive excess at $s$. In case (c), although $M_s \kappa_s$ increases, $M_t \kappa_t$ decreases by a larger amount. $\square$

The dual description of an optimal solution is in terms of *relabelings* with a label function $\mu : V \to \mathbb{R}_{>0} \cup \{\infty\}$. For each node $i \in V$, let us rescale the flow on all arcs $ij \in E$ by $\mu_i$: let $f^\mu_{ij} = f_{ij}/\mu_i$. We get a problem equivalent to the original one with relabeled gains $\gamma^\mu_{ij} = \gamma_{ij}\mu_i/\mu_j$. Accordingly, the relabeled demands and excesses are $b^\mu_i = b_i/\mu_i$, $e^\mu_i = e_i/\mu_i$. A relabeling is *conservative*, if for any residual arc $ij \in E_f$, $\gamma^\mu_{ij} \leq 1$, that is, no arc may increase the relabeled flow. Furthermore, for each $i \in V$, $\mu_i \geq 1/M_i$ is required and that equality must hold whenever $e_i < 0$.

We use the conventions $\infty \cdot 0 = 0$ and $\infty/\infty = 0$. Accordingly, if $\mu_i = \infty$, we define $b^\mu_i = e^\mu_i = 0$, and $\gamma^\mu_{ji} = 0$ for all arcs $ji \in E_f$. If $ij \in E_f$, $\mu_i = \infty$ and $\mu_j < \infty$, then $\gamma^\mu_{ij} = \infty$. Consequently, if $\mu$ is conservative, then $\mu_i < \infty$ for any $i \in V$ for which there exists a path in $E_f$ from $i$ to any node $t \in V$ with $e_t < 0$. Also, if $\mu$ is conservative, there exists no flow generating cycle on the node set $\{i : \mu_i < \infty\}$. This is since for a cycle $C$, $\gamma(C) = \Pi_{ij\in C}\gamma_{ij} = \Pi_{ij\in C}\gamma^\mu_{ij}$.

**Theorem 5.2.** *For a pseudoflow $f$, the following are equivalent.*

*(i) $f$ is an optimal solution to the symmetric generalized flow problem.*

*(ii) $E_f$ contains no generalized augmenting paths.*

*(iii) There exists a conservative relabeling $\mu$ with $e_i = 0$ whenever $1/M_i < \mu_i < \infty$.*

*Proof.* The equivalence of (i) and (iii) is by linear programming duality, with $\mu_i$ being the reciprocal of the dual variable corresponding to the inequality $e_i + \kappa_i \geq 0$. (i) implies (ii) by Lemma 5.1.

It is left to show that (ii) implies (iii). If the excess discrepancy is 0 (that is, $e_i \geq 0$ for all $i \in V$), then $\mu \equiv \infty$ is conservative. Otherwise, let $N = \{t : e_t < 0\}$. If $E_f$ contains no directed path from $i \in V$ to $N$, then let $\mu_i = \infty$. For the other nodes $i \in V$, let $\mu_i$ be the smallest possible value of $1/(\gamma(P)M_t)$ for $\gamma(P) = \Pi_{e\in P}\gamma_e$, where $P$ is a walk in $E_f$ starting from $i$ and ending in a node $t \in N$. By (ii), this is well-defined, since all cycles can be removed from a walk $P$ without decreasing $\gamma(P)$.

$\mu$ clearly satisfies $\gamma_{ij}\mu_i/\mu_j \leq 1$. We shall prove $\mu_i \geq 1/M_i$ for each $i \in V$ and $\mu_t = 1/M_t$ for each $t \in N$. The first claim is trivial, since the gain of the path $P = \emptyset$ is defined as 1. If $\mu_t > 1/M_t$, then there exists a path $P$ from $t$ to some $t' \in N$ with $1/(\gamma(P)M_{t'}) < 1/M_t$, giving a type (c) GAP, a contradiction. Finally, if $e_s > 0$ for a node $s$ with $\mu_s < \infty$, then there would exists a GAP of type (b). □

## 5.2 Canonical labels

Given a pseudoflow $f$ and a conservative labeling $\mu$, the arc $ij \in E_f$ is called *tight* if $\gamma^\mu_{ij} = 1$. A directed path in $E_f$ is called *tight* if it consists of tight arcs.

We introduce arc imbalances as in [15]. This can be done in a simplified manner because of using an uncapacitated network; in particular, we need imbalances only for reverse arcs. (The transformation will be given in Section 5.4).

For every arc $ij \in E$, $\bar{e}_{ij} \geq 0$ will denote its *arc imbalance*, which accounts for some leftover flow in node $j$. $\bar{e}_{ij}$ is defined and can be positive even if $ji \notin E_f$. We introduce the *modified excess* $\bar{e}_i$ for each $i \in V$. These shall satisfy

$$e_i = \bar{e}_i + \sum_{ji\in E} \bar{e}_{ji} \quad \forall i \in V \tag{4}$$

We apply relabeling for the $\bar{e}_i$'s and $\bar{e}_{ij}$'s as well, with $\bar{e}^\mu_i = \bar{e}_i/\mu_i$, $\bar{e}^\mu_{ij} = \bar{e}_{ij}/\mu_j$. We will always maintain $0 \leq \bar{e}^\mu_{ij} < \Delta$ and $\bar{e}_{ij} = 0$ whenever $e_j < 0$.

A node $i \in V$ is called *negative* if $\bar{e}_i < 0$, *neutral* if $\bar{e}_i = 0$ and *positive* if $\bar{e}_i > 0$. If all nodes are nonnegative, then $e_i \geq 0$ for each $i \in V$ and hence the current solution is optimal with excess discrepancy 0. In the sequel, assume some negative nodes exist. For a relabeling $\mu$, let $Ex^\mu(f) = \sum_{i\in V-N} \bar{e}_i^\mu$ denote the total positive excess.

The relabeling $\mu$ is called *canonical*, if it is conservative, and for each $i \in V$ with $\mu_i < \infty$, there exists a tight path in $E_f$ from $i$ to some negative node. Given a relabeling $\mu$ which is not canonical, the subroutine Tighten-Label$(f, \mu)$ replaces $\mu$ by a canonical labeling $\mu'$ with $\mu'_i \geq \mu_i$ for each $i \in V$.

Tighten-Label$(f, \mu)$ is essentialy Dijkstra's algorithm. Let $V' \subseteq V$ be the set of nodes $i$ with a directed path in $E_f$ from $i$ to a negative node. For nodes in $V - V'$, let us set $\mu_i = \infty$. Let $S \subseteq V'$ be the set of nodes $i$ for which there exists a (possibly empty) tight path for the current $\mu$ to a negative node. In each step of the algorithm, $S$ will be extended by at least one element, and we terminate if $S = V'$, when the current relabeling is canonical.

If $V' - S \neq \emptyset$, let us multiply $\mu_i$ for each $i \in V' - S$ by $\alpha$ defined as

$$\alpha = \min\left\{\frac{1}{\gamma_{ij}^\mu} : ij \in E_f, i \in V' - S, j \in S\right\}.$$

By the definition of $S$, $\alpha > 1$, and after multiplying by $\alpha$, at least one arc $ij \in E_f$ with $i \in V' - S$, $j \in S$ will become tight. Tight arcs inside $S$ also remain tight, hence $S$ is extended by at most one node. Also, the choice of $\alpha$ guarantees that $\mu$ remains conservative.

### 5.3 Description of the algorithm

---

**Algorithm** Symmetric-Generalized-Flows
*Transform to uncapacitated network with* $n' = n + m$ *nodes and* $m' = 2m$ *arcs;*
*Find initial pseudoflow* $f^0$ *and conservative labeling* $\mu^0$;
$f \leftarrow f^0$, $\mu \leftarrow \mu^0$;
**while** $Ex^\mu(f) \geq 1/B^{n'+m'}$ **do**
    **for** $ij \in E$ **do** $\bar{e}_{ij} \leftarrow 0$;
    **for** $i \in V$ **do** $\bar{e}_i \leftarrow e_i$;
    $\Delta \leftarrow Ex^\mu(f)/(2n' + 2m')$;
    **do**
        Tighten-Label$(\mu, f)$;
        $D \leftarrow \{i \in V : \bar{e}_i \geq \Delta\}$;
        $N \leftarrow \{i \in V : \bar{e}_i < 0\}$;
        **if** $N = \emptyset$ **then return** optimal primal solution $f$; **terminate**;
        **pick** $s \in D$, $t \in N$ connected by a tight path $P$;
        Push$(P)$;
    **while** $D \neq \emptyset$;
**compute** a maximum flow from nodes $\{s \in V : e_s > 0, \mu_s < \infty\}$
            to nodes $\{t \in V : e_t < 0\}$ using tight arcs for $\mu$;
**return** optimal primal solution $f$ and optimal dual solution $\mu$

---

Figure 1: The algorithm for symmetric generalized flows

An outline of the algorithm is given in Figure 1. First, we transform the problem to an equivalent one with all lower arc capacities 0 and no upper arc capacities. This increases the number of nodes to $n' = n + m$ and the number of arcs to $2m$. The transformation is described in Section 5.4. There we shall also define an initial solution $f_0$ along with a conservative labeling $\mu_0$. In the sequel, assume we have such an uncapacitated network and the initial solution.

During the algorithm we always maintain a pseudoflow $f$ along with a conservative labeling $\mu$. The $\mu_i$ values can only increase, starting from $\mu \equiv \mu^0$. Let $N$ denote the set of negative nodes. During the algorithm, no new elements enter $N$, and for each $i \in N$, $\mu_i = 1/M_i$ is maintained. If $N = \emptyset$ at some point, the algorithm terminates with an optimal (0-discrepancy) solution.

The algorithm consists of $\Delta$-scaling phases. At the beginning of each $\Delta$-phase, we set $\bar{e}_i = e_i$ for all $i \in V$, and reset the arc balances $\bar{e}_{ij} = 0$ for all $ij \in E$. We also update $\mu$ to a canonical labeling by calling Tighten-Label$(f, \mu)$. If $Ex^\mu(f) < 1/B^{n+3m}$, then an optimal solution can be found by a single maximum flow computation, as it shall be shown in Section 5.5. In this case we terminate.

Otherwise, we set the new value $\Delta = Ex^\mu(f)/(2n' + 2m')$. We shall show that $\Delta$ decreases at least by a factor of two compared to the previous phase.

In the first phase, the initial $\mu \equiv \mu^0$ is conservative for $f = f^0$. In later phases we always start with a conservative labeling from the previous phase. Let $D$ denote the set of nodes $i$ with $\bar{e}_i^\mu \geq \Delta$. The $\Delta$-phase consists of a sequence of iterations, and terminates whenever $D$ becomes empty.

At the beginning of an iteration, we update $\mu$ to a canonical labeling by calling Tighten-Label$(f, \mu)$. Note that this might only decrease the positive excesses $\bar{e}_i^\mu > 0$, hence it is possible that $D$ becomes empty. If $D \neq \emptyset$, we pick an arbitrary $s \in D$, and try to push $\Delta$ units of flow from $s$ to some $t \in N$ on a tight path $P$ using the subroutine Push$(P)$. Then we move to the next iteration.

**Subroutine Push**$(P)$

Let $P = i_0 i_1 \ldots i_k$ with $i_0 = s$ and $i_k = t$. We push flow along $P$ arc-by-arc. Let $0 \leq h \leq k$ be the current step. If $h = k$ or $\bar{e}_{i_h}^\mu < \Delta$, then Push$(P)$ terminates.

Assume now $\bar{e}_{i_h}^\mu \geq \Delta$. First, consider the case when the residual capacity of $i_h i_{h+1}$ is at least $\Delta$. (that is, $i_h i_{h+1}$ is a forward arc, or it is a backward arc with $f_{i_{h+1} i_h}^\mu \geq \Delta$). Let us decrease the relabeled excess $\bar{e}_{i_h}^\mu$ by $\Delta$, increase $f_{i_h i_{h+1}}^\mu$ and $\bar{e}_{i_{h+1}}^\mu$ by $\Delta$, and proceed to the next step.

Assume now that $i_h i_{h+1}$ is backward arc with residual capacity $\alpha < \Delta$. Then decrease $\bar{e}_{i_h}^\mu$ by $\Delta$, increase $\bar{e}_{i_{h+1}}^\mu$ by $\alpha$, and saturate the arc $i_h i_{h+1}$ (since it was a backward arc, this means setting $f_{i_{h+1} i_h}^\mu$ from $\alpha$ to zero). Furthermore, increase the arc imbalance $\bar{e}_{i_{h+1} i_h}^\mu$ by $\Delta - \alpha$.

## 5.4 Transformation to an uncapacitated problem

In this section, we show how the problem is transformed to an uncapacitated version, and define the initial pseudoflow $f^0$ and initial relabeling $\mu^0$. Such a transformation is a standard technique in the flow literature.

Let the node set $V'$ consist of the original node set $V$ and a new node corresponding to each arc; thus the number of nodes is $n' = n + m$. The original nodes are called *primary* nodes, and those corresponding to arcs *secondary* nodes. Let $k = a_{ij}$ be the node corresponding to $ij$. The transformed graph contains two corresponding arcs, $ik$ and $jk$. Thus the number of arcs is $m' = 2m$. We will denote the modified data by $b'$, $M'$, $\gamma'$. In the other sections, we use the original notation for the transformed instance for simplicity.

Let us set $\ell' \equiv 0$ and $u' \equiv \infty$ as capacities. For a primary node $i \in V'$, let us set the node demand $b'_i = b_i - \sum_{j:ji\in E} \gamma_{ji} u_{ji} + \sum_{j:ij\in E} \ell_{ij}$ and let $M'_i = M_i$.

For the secondary node $k = a_{ij}$, let $b'_k = \gamma_{ij}(u_{ij} - \ell_{ij})$ and $M'_k = M_j$. Furthermore, let us define the gain factors by $\gamma'_{ik} = \gamma_{ij}$, $\gamma'_{kj} = 1$.

**Lemma 5.3.** *(i) A pseudoflow $f$ in the original problem can be transformed to a pseudoflow $f'$ in the modified problem with the same excess discrepancy. If $f'$ is a pseudoflow in the modified problem, then it can be transformed to a pseudoflow for the original problem with the same or smaller discrepancy.*

*(ii) All arc gains can be represented by quotients of integers at most $B$, all node demands are integer multipliers of $1/B^*$ for some $B^* \le B^{n+3m}$, and their absolute value is at most $B(n+1)$.*

*Proof.* For (i), let $f$ be a pseudoflow in the original problem. For $k = a_{ij}$, let $f'_{ik} = f_{ij} - \ell_{ij}$ and $f'_{jk} = \gamma_{ij}(u_{ij} - f_{ij})$. Then $e'_k = 0$, and $e'_i = e_i$ for all primary nodes.

Let us now take a pseudoflow $f'$ in the transformed problem, and let $k = a_{ij}$. If $f'_{jk} \ge e'_k$, then let us decrease $f'_{jk}$ by $e'_k$. Since $M'_k = M'_j$ and $\gamma'_{jk} = 1$, this cannot increase the excess discrepancy. If $f'_{jk} < e'_k$, then $f'_{ik} > e'_k > 0$ follows. Decreasing $f'_{ik}$ by $e'_k/\gamma'_{ik}$ can also only decrease the excess discrepancy.

If $e'_k = 0$ for all secondary nodes $k$, then we can define $f_{ij} = f'_{ik} + \ell_{ij}$ for $k = a_{ij}$. This will be a pseudoflow in the original instance with $e'_i = e_i$ for all $i \in V$.

Let us now turn to the proof of (ii). Before the transformation, all arc gains and node demands were quotients of two integers, each at most $B$. This holds for the arc new gains as well. All new node demands will have a common denominator $B^* \le B^{n+3m}$, the common denominator of all original $b_i$'s, $\gamma_{ij} u_{ij}$'s and $\ell_{ij}$'s.

The absolute value of the new node demands is at most $B(n+1)$, assuming the original graph is simple. □

Let us now define the initial solution $f^0$ and conservative labeling $\mu^0$. Let us set $\mu^0_i = 1/M'_i$ for all primary nodes $i$. Let $k = a_{ij}$ be a secondary node. If $\gamma'_{ik} > M'_i/M'_j$, let us set $f^0_{ik} = b'_k$ and $\mu^0_k = \gamma'_{ik}/M'_i$. If $\gamma'_{ik} \le M'_i/M'_j$, set $f^0_{ik} = 0$ and $\mu^0_k = 1/M'_k = 1/M'_j$. In both cases, let $f'_{jk} = 0$. It is easy to see that this is a conservative labelling with $\mu^0_i \ge 1/M'_i$ for all nodes $i$, and $\mu^0_i = 1/M'_i$ whenever $e'_i < 0$.

## 5.5 Moving to an optimal solution

In this section, we assume all arc imbalances are 0 and thus $e^\mu_i = \bar{e}^\mu_i$ for all nodes $i \in V$. We shall show that if $Ex^\mu(f) < 1/B^{n+3m}$, then a single maximum flow computation yields an optimal solution. This is the standard technique how most algorithms in the literature terminate.

**Lemma 5.4.** *Let $\mu$ be a canonical relabeling for $f$, and let $\tilde{G}_f = (V, \tilde{E}_f)$ be the subgraph of $G_f$ consisting of tight arcs in $E_f$. If $Ex^\mu(f) < 1/B^{n+3m}$, then a single maximum flow computation on $\tilde{G}$ from source set $P = \{s \in V : e_s > 0, \mu_s < \infty\}$ to sink set $N = \{t \in V : e_t < 0\}$ terminates with an optimal solution.*

*Proof.* Consider the flow $f'$ resulting after the maximum flow computation. Since flow was sent only on tight arcs, $\mu$ is also conservative for $f'$. If there are no more nodes $s$ with $e^\mu_s > 0$, then by Theorem 5.2, $f'$ is optimal. Assume now $P'$, the set of such nodes for $f'$ is nonempty.

Let $S \subseteq V$ be the set of nodes reachable from $P'$ using tight residual arcs in $E_{f'}$. By optimality, $S$ contains no node with negative excess. No backward arc in $E_{f'}$ may leave or enter $S$, since in a conservative labeling, all backward arcs (along with their reverse arcs) are tight; also, all $\mu_i$'s in $S$ are finite. Consider now a forward arc $ij \in E_{f'}$ leaving $S$. By definition, $ij$ cannot be tight. This implies $f'_{ij} = 0$ (equivalently, $ji \notin E_{f'}$). Similarly, if a forward arc $ij \in E_{f'}$ enters $S$, then $f'_{ij} = 0$ follows.

Hence no flow in $f'$ enters or leaves $S$. Also, on all arcs $ij$ with $f'_{ij} > 0$, $\gamma^{\mu}_{ij} = 1$ because of conservativity. Therefore,

$$0 \le \sum_{i \in S} e^{\mu}_i(f') = \sum_{i \in S} \left( \sum_{j:ji \in E} \gamma^{\mu}_{ji} {f'_{ji}}^{\mu} - \sum_{j:ij \in E} {f'_{ij}}^{\mu} - b^{\mu}_i \right) = -\sum_{i \in S} b^{\mu}_i.$$

By the assumption of the theorem, $\sum_{i \in S} e^{\mu}_i(f') \le Ex^{\mu}_f \le 1/B^{n+3m}$, hence $0 \le -\sum_{i \in S} b^{\mu}_i \le 1/B^{n+3m}$.

Since $\mu$ is a canonical labeling for $f$, there exists a minimal set $F \subseteq E_f$ of tight arcs for $f$ that contains a tight path from each node $i$ to a negative node $t$. $|F| \le n' = n + m$, since at most one arc leaves every node. $1/\mu_i$ is then the product of $M_t$ and the gain factors on such a tight path. Hence a common denominator of all $\gamma_{ij}$'s and for $ij \in F$ is also a common denominator for all $1/\mu_i$'s. Using Lemma 5.3, there is a common denominator $B^*$ of the rational numbers in $\{\gamma_{ij} : ij \in F\}$ and of all $b_i$'s with $B^* \le B^{n+3m}$. Therefore $B^*$ is also common denominator of all rational numbers $b^{\mu}_i = b_i/\mu_i$. Consequently, if $0 \le -\sum_{i \in S} b^{\mu}_i \le 1/B^{n+3m}$ implies $\sum_{i \in S} e^{\mu}_i(f') = -\sum_{i \in S} b^{\mu}_i = 0$, proving the theorem. $\square$

### 5.6 Analysis

First, observe that conservativeness is maintained all the time, since in Push$(P)$, we increase $f_{ij}$ only on tight arcs.

**Claim 5.5.** *During the entire $\Delta$-phase, (4) is maintained and $e^{\mu}_{ij} < \Delta$ for all arcs $ij \in E$.*

*Proof.* The first claim is straigtforward. We prove the second claim by induction along with $\bar{e}^{\mu}_{ij} < f^{\mu}_{ij}$ whenever $f^{\mu}_{ij} > 0$ (that is, if $ji \in E_f$, $\mu_j < \infty$).

Initially, all $\bar{e}^{\mu}_{ij}$'s are 0. Note that Tighten-Label$(f, \mu)$ may only decrease the value of $\bar{e}^{\mu}_{ij}$. Each time $\bar{e}^{\mu}_{ij}$ is increased, $f^{\mu}_{ij}$ is set to 0. Between two increases of $\bar{e}^{\mu}_{ij}$, $f^{\mu}_{ij}$ has to be increased. Since $ij \in E$ is a forward arc, $f^{\mu}_{ij}$ increases by exactly $\Delta$. By induction, we have $\bar{e}^{\mu}_{ij} < \Delta = f^{\mu}_{ij}$ in this step. $\bar{e}^{\mu}_{ij} < f^{\mu}_{ij}$ is maintained by further increases of $f^{\mu}_{ij}$ and by all subsequent relabeling steps. Let us verify the latter claim. since $ji \in E_f$, both arcs $ij$ and $ji$ are tight for $\mu$. Hence Tighten-Label$(f, \mu)$ always increases $i$ and $j$ by the same factor. (Note that $\bar{e}^{\mu}_{ij} = e_{ij}/\mu_j$, while $f^{\mu}_{ij} = f_{ij}/\mu_i$.)

Consider now the next step when $\bar{e}^{\mu}_{ij}$ increases. This can happen if $f^{\mu}_{ij} < \Delta$. Having had $\bar{e}^{\mu}_{ij} < f^{\mu}_{ij}$ before this step, and since $\bar{e}^{\mu}_{ij}$ increases by $\Delta - f^{\mu}_{ij}$, the new value of $\bar{e}^{\mu}_{ij}$ will be strictly less than $\Delta$. $\square$

At the termination of the $\Delta$-phase, the above claim and the condition $D = \emptyset$ immediately yields $Ex^{\mu}(f) \le n'\Delta$, and after resetting all arc imbalances to 0, we still have $Ex^{\mu}(f) \le (n' + m')\Delta$. The next lemma then follows.

**Lemma 5.6.** *The value of $\Delta$ decreases by a factor of at least two between two phases.* $\square$

**Lemma 5.7.** *A $\Delta$-phase calls* Push *at most $2(n' + m') = 2n + 6m$ times.*

*Proof.* Consider the potential function $\Psi = \sum_{i \in V - N} \lfloor \bar{e}^{\mu}_i / \Delta \rfloor$. By the choice of $\Delta$, $\Psi \le 2(n' + m')$ at the beginning. In the relabeling steps, $\Psi$ may only decrease. Otherwise, it can be modified during the Push steps. When moving from $i_h$ to $i_{h+1}$, $\bar{e}^{\mu}_{i_h}$ decreases by $\Delta$ and $\bar{e}^{\mu}_{i_{h+1}}$ increases by at most $\Delta$, hence $\Psi$ cannot increase. If we terminate in $i_{h+1}$, then $\Psi$ should decrease by at least one, either since $i_{h+1} \in N$ or because both the old and the new value of $\bar{e}^{\mu}_{i_{h+1}}$ are strictly less than $\Delta$. $\square$

**Theorem 5.8.** *The algorithm runs in $O(m^2(m \log B + \log M) \log m)$ time.*

*Proof.* By Lemma 5.3 and the definition of $f^0$, the initial value $Ex^\mu(f^0)$ is at most the sum of $-b_i/\mu_i$'s on primary nodes, hence at most $MBn(n+1)$. By Lemma 5.6 and the termination criteria $Ex_f^\mu < 1/B^{n+3m}$, the total number of phases is upper bounded by $O(m \log B + \log M)$.

The number of iterations is $O(m)$ by Lemma 5.7. The running time of an iteration is dominated by the TIGHTEN-LABEL step, that can be done in $O(m' + n' \log n') = O(m \log m)$ time using Fredman and Tarjan's [10] implementation of Dijkstra's algorithm. □

# 6 Concave generalized flows algorithm

## 6.1 Comparison to the generalized flow setting

The algorithm for the concave problem is mostly analogous to the one for generalized flows described in the previous section, yet there are remarkable differences. In particular, this algorithm specialized for linear gain functions is different from the one presented in the previous section. We start with exhibiting the main differences.

- For generalized flows, we aimed to find an optimal solution, now we only want to find a $\varepsilon$-approximate solution for some $\varepsilon > 0$. For generalized flows, Lemma 5.4 obtained an optimal solution by a single max flow computation when $\Delta < B^{n+3m}/(2n+6m)$; for the concave setting, $\Delta < \varepsilon/(2n+8m)$ guarantees $\varepsilon$-optimality by Theorem 6.14.

- We shall also transform the concave problem to an uncapacitated one (Section 6.2). This is essentially different from the transformation of Section 5.4: there we added a new node for each arc. Here we keep the original network, and only modify the gain functions by encoding the explicit upper bounds. For generalized flows, this gives nonlinear gain functions. Nevertheless, on each arc $ij$ we maintain the implicit upper bound $\tilde{u}_{ij}$ so that the function $\Gamma_{ij}$ is constant on $[\tilde{u}_{ij}, \infty]$.

- In the $\Delta$-phase, we use the same residual network $E_f$ as for generalized flows, and we linearize the gain functions in chunks $\Delta$, similarly as (3) for convex minimum-cost flows. Hence when moving from phase $\Delta$ to $\Delta/2$, some arcs might violate the relaxed dual optimality condition, called $\Delta$-conservativness. In this case, the new subroutine ADJUST($\Delta$) will modify each relabeled $f_{ij}^\mu$ by at most $\Delta$. Because of this subroutine, we always decrease $\Delta$ precisely by a factor of 2, while for generalized flows, a larger decrease was also possible.

  ADJUST($\Delta$) yields $\Delta/2$-conservativeness, but causes a serious new difficulty, since the $e_i^\mu$ values can decrease by at most $d_i\Delta$. This could turn some nodes with $e_i > 0$ to $e_i < 0$, and thus it would be impossible to maintain $\mu_i = 1/M_i$ for all $e_i < 0$.

- To disable ADJUST($\Delta$) creating new nodes with $e_i < 0$, $\mu_i > 1/M_i$, we modify the definition of positive and negative nodes by reserving a part of the excess to account for later modifications. For simplicity, assume all arc imbalances are zero. In the generalized flow setting, an arc was called negative if $e_i < 0$; in contrast, here we shall call $i$ negative if $e_i^\mu < d_i\Delta$; we maintain $\mu_i = 1/M_i$ for all negative nodes. Hence ADJUST($\Delta$) cannot create any new negative nodes. As a side effect, we will also push flow to nodes with $0 \le e_i < d_i\Delta$, but as $\Delta$ decreases, such nodes will turn positive and thus we remove their excess.

- Due to the modified definition of positive and negative nodes, the definition and subroutine for $\Delta$-canonical labelings also has to be changed. Increasing $\mu_i$ for a positive node $i$ could turn it into negative. We avoid it by stopping to increase $\mu_i$ whenever $i$ becomes neutral, that is, $e_i^\mu = d_i\Delta$. For this reason, in a $\Delta$-canonical relabeling we cannot maintain a tight path from each nonnegative node to a negative one, but only from positive nodes to nonnegative ones. Hence in the algorithm, we will occasionaly push flow to neutral nodes. Still, we can maintain $\bar{e}_i^\mu \leq \Delta$ at the end of a $\Delta$-phase (for generalized flows, we had strict inequality).

### 6.2 Transformation to an uncapacitated problem

In the concave case, we shall also transform the problem to an equivalent version with $\ell \equiv 0$ and no upper bounds. This transformation will be essentialy different from the one in Section 5.4: the number of nodes and arcs are left unchanged. The explicit upper bounds are removed, yet they are encoded into the gain functions.

First, we transform the problem to an equivalent version with $\ell \equiv 0$ and $\Gamma_{ij}(0) = 0$ whenever $\Gamma_{ij}(0) > \infty$. On each arc $ij \in E$, let us replace $u_{ij}$ by $u_{ij} - \ell_{ij}$ and $\Gamma_{ij}(\alpha)$ by $\Gamma_{ij}(\alpha + \ell_{ij}) - \Gamma_{ij}(\ell_{ij})$. On each node $i \in V$, let us increase $b_i$ by $\sum_{ij \in E} \ell_{ij}$, and decrease it by $\Gamma_{ji}(\ell_{ji})$ for all arcs $ji \in E$ where this value is finite.

Next, we remove the upper bounds by encoding them into the $\Gamma_{ij}$'s. Let us extend each function $\Gamma_{ij}$ from $[0, u_{ij}]$ to $\mathbb{R}_+$ by setting $\Gamma_{ij}(p) = \Gamma_{ij}(u_{ij})$ for arbitrary $p > u_{ij}$. This preserves concavity.

Also, let us define $\tilde{u}_{ij} = \inf\{p : p \geq 0, \Gamma_{ij}(p) = \Gamma_{ij}(u_{ij})\}$. By concavity, $\Gamma(\tilde{u}_{ij}) = \Gamma_{ij}(u_{ij})$, and $\Gamma_{ij}(u_{ij})$ is strictly monotone increasing on the interval $[0, \tilde{u}_{ij}]$. Note that $\tilde{u}_{ij} \leq U$.

The transformation increases $U$ by at most a factor of $n + 1$, assuming the graph is simple. Since in the complexity estimation we have $U$ only in the term $\log(MUm/\varepsilon)$, this change can be ignored. Unlike for generalized flows, we will not need the original input problem anymore and work exclusively with the transformed instance.

### 6.3 Optimality conditions

The concavity of $\Gamma_{ij}$ implies that for each $0 \leq \alpha$, there exists the right derivative, denoted by $\Gamma_{ij}^+(\alpha)$, and for $0 < \alpha$, there exists the left derivative $\Gamma_{ij}^-(\alpha)$. If $0 < \Delta < \Delta'$, then

$$\frac{\Gamma_{ij}(\alpha + \Delta') - \Gamma_{ij}(\alpha)}{\Delta'} \leq \frac{\Gamma_{ij}(\alpha + \Delta) - \Gamma_{ij}(\alpha)}{\Delta} \leq \Gamma_{ij}^+(\alpha), \tag{5a}$$

$$\frac{\Gamma_{ij}(\alpha) - \Gamma_{ij}(\alpha - \Delta')}{\Delta'} \geq \frac{\Gamma_{ij}(\alpha) - \Gamma_{ij}(\alpha - \Delta)}{\Delta} \geq \Gamma_{ij}^-(\alpha) \tag{5b}$$

for $0 \leq \alpha$ and for $\Delta' \leq \alpha$, respectively. Furthermore, if $0 \leq \alpha < \alpha'$, then $\Gamma_{ij}^+(\alpha') \leq \Gamma_{ij}^-(\alpha') \leq \Gamma_{ij}^+(\alpha) \leq \Gamma_{ij}^-(\alpha)$. Notice also that $\Gamma_{ij}^+(\alpha) = 0$ if $\alpha \geq \tilde{u}_{ij}$ and $\Gamma_{ij}^-(\alpha) = 0$ if $\alpha > \tilde{u}_{ij}$.

For a pseudoflow $f : E \to \mathbb{R}$, we define the residual network $G_f = (V, E_f)$ identical as for the generalized flow setting: $ij \in E_f$ if $ij \in E$ or $ji \in E$ and $f_{ji} > 0$. For notational convenience, we define $f_{ji} = -\Gamma_{ij}(f_{ij})$ on backward arcs. We also define the function $\Gamma_{ji}(\alpha) : [-\Gamma_{ij}(\tilde{u}_{ij}), \Gamma_{ij}(0)] \to [-\tilde{u}_{ij}, 0]$ by

$$\Gamma_{ji}(\alpha) = -\Gamma_{ij}^{-1}(-\alpha).$$

(the inverse is defined with $\Gamma_{ij}^{-1}(\Gamma_{ij}(\tilde{u}_{ij})) = \tilde{u}_{ij}$). Hence $\Gamma_{ji}(f_{ji}) = -f_{ij}$. The following claim is also easy to verify.

**Claim 6.1.** *For any $ij \in E$ with $0 < f_{ij} \le \tilde{u}_{ij}$, $\Gamma^+_{ij}(f_{ij}) = 1/\Gamma^-_{ji}(f_{ji})$, $\Gamma^-_{ij}(f_{ij}) = 1/\Gamma^+_{ji}(f_{ji})$.* □

In the rest of the section, we assume $f \le \tilde{u}$. Note that on arcs with $f_{ij} > \tilde{u}_{ij}$, we can decrease to $f_{ij} = \tilde{u}_{ij}$ without increasing the excess discrepancy $\kappa_f$.

Let $P = i_0 \dots i_k$ be a walk in the auxiliary graph $E_f$. By sending $\alpha$ units of flow along $P$, we mean the following. First we increase $f_{i_0 i_1}$ by $\alpha$ and set $\beta = \Gamma_{i_0 i_1}(f_{i_0 i_1} + \alpha) - \Gamma_{i_0 i_1}(f_{i_0 i_1})$ to be the flow arriving at $i_1$. In step $h = 1, \dots, k-1$, we increase the flow on $i_h i_{h+1}$ by $\beta$ and set the new value of $\beta$ as $\Gamma_{i_h i_{h+1}}(f_{i_h i_{h+1}} + \beta) - \Gamma_{i_h i_{h+1}}(f_{i_h i_{h+1}})$. We assume $\alpha$ is chosen small enough so that no capacity gets violated. Let $f^{\alpha,P}$ denote the modified flow.

If $C = i_0 \dots i_{k-1}$ is a cycle if $E_f$, then by sending $\alpha$ units of flow around $C$ from $i_0$ we mean sending $\alpha$ units on the walk $i_0 \dots i_{k-1} i_0$. This modifies $e_i$ only in node $i_0$: if the flow increase from $i_{k-1} i_0$ is bigger than $\alpha$, then $e_{i_0}$ increases, and if it is smaller then it decreases. The next lemma characterizes when $e_{i_0}$ can increase.

**Lemma 6.2.** *Let $C$ be a cycle in $E_f$ with $i_0 \in V(C)$. If $\Gamma^+_f(C) > 1$ then $e_{i_0}$ can be increased by sending some flow around $C$. If $\Gamma^+_f(C) \le 1$, then it is not possible to increase $e_{i_0}$ by sending any $\alpha > 0$ amount around $C$.* □

Since this property is independent from the choice of $i_0$, we simply say that $C$ is a *flow generating cycle* if $\Gamma^+_f(C) > 1$. The lemma is an immediate consequence of the following claim.

**Claim 6.3.** *For a walk $P = \{i_0 i_1 \dots i_k\}$ in $E_f$, let $\Gamma^+_f(P) = \Pi_{ij \in C} \Gamma^+_{ij}(f_{ij})$. For any value of $\alpha > 0$, the flow increase in $i_k$ for $f^{\alpha,P}$ is at most $\Gamma^+_f(P)\alpha$. On the other hand, for any $\varepsilon > 0$ there exists a $\delta > 0$ so that for any $0 < \alpha \le \delta$, $f^{\alpha,P}$ increases $e_{i_k}$ by at least $(\Gamma^+_f(P) - \varepsilon)\alpha$.*

*Proof.* The first part is trivial by concavity. We prove the second part by induction on the subpaths $P_h = \{i_0 \dots i_h\}$ for $h = 0, \dots, k$. We may assume $\Gamma^+_f(P) > 0$, as $e_{i_k}$ is left unchanged if $\Gamma^+_{ij}(f_{ij}) = 0$ on some arc. There is nothing to prove for $h = 0$; assume we have already proved it for $P_{h-1}$. By the definition of $\Gamma^+_{i_{h-1} i_h}$, for each $\varepsilon^* > 0$ there exists a $\delta^* > 0$ such that for any $0 < \beta \le \delta^*$,

$$(\Gamma^+_{i_{h-1} i_h}(f_{i_{h-1} i_h}) - \varepsilon^*)\beta \le \Gamma_{i_{h-1} i_h}(f_{i_{h-1} i_h} + \beta) - \Gamma_{i_{h-1} i_h}(f_{i_{h-1} i_h}).$$

Choose now an $\varepsilon > 0$. From the induction hypothesis, we can choose a small enough $\varepsilon^* > 0$ such that

$$\Gamma^+_f(P_h) - \varepsilon < (\Gamma^+_{i_{h-1} i_h}(f_{i_{h-1} i_h}) - \varepsilon^*)(\Gamma^+_f(P_{h-1}) - \varepsilon^*).$$

Let us choose an appropriate $\delta^*$ for $i_{h-1} i_h$. By induction, we can choose a small enough $\delta > 0$ with the following properties: If $0 < \alpha < \delta$, then $\Gamma^+_f(P_{h-1})\alpha \le \delta^*$, and the increase of $e_{h-1}$ for $f^{\alpha, P_{h-1}}$ is at least $\beta = (\Gamma^+_f(P_{h-1}) - \varepsilon^*)\alpha$. □

The definition of GAPs is identical as for generalized flows, with the single difference in case (c), where $\Gamma^+_f(P) > M_s/M_t$ replaces $\gamma(P) > M_s/M_t$. The following lemma can also be proved identical as Lemma 5.1.

**Lemma 6.4.** *If $f$ is an optimal solution, then no GAP exists.* □

*Relabelings* are also defined analogously as for generalized flows. Given $\mu : V \to \mathbb{R}_{>0} \cup \{\infty\}$, let us define $f^\mu_{ij} = f_{ij}/\mu_i$ for each arc $ij \in E$. We get problems equivalent to the original with relabeled functions $\Gamma^\mu_{ij}(\alpha) = \Gamma_{ij}(\mu_i \alpha)/\mu_j$. Accordingly, the relabeled demands and excesses are $b^\mu_i = b_i/\mu_i$,

$e_i^\mu = e_i/\mu_i$, and $\tilde{u}_{ij}^\mu = \tilde{u}_{ij}/\mu_i$. A relabeling is *conservative*, if for any residual arc $ij \in E_f$, $\Gamma_{ij}^{\mu+}(f_{ij}^\mu) \le 1$, that is, no edge may increase the relabeled flow. Furthermore we require $\mu_i \ge 1/M_i$ for every $i \in V$ and equality whenever $e_i < 0$.

We use the same convention for infinite $\mu_i$ values as for generalized flows. If $\mu_i = \infty$, we define $b_i^\mu = e_i^\mu = 0$, $\tilde{u}_{ij}^\mu = 0$ for $ij \in E$, and furthermore $\Gamma_{ji}^{\mu+}(f_{ji}^\mu) = 0$ for all arcs $ji \in E_f$. Finally, consider an arc $ij \in E_f$ with $\mu_i = \infty$, $\mu_j < \infty$. If $f_{ij} < \tilde{u}_{ij}$, let $\Gamma_{ij}^{\mu+}(f_{ij}^\mu) = \infty$; if $f_{ij} \ge \tilde{u}_{ij}$, let $\Gamma_{ij}^{\mu+}(f_{ij}^\mu) = 0$.

If $\mu$ is conservative, then if for a node $i \in V$ there exists a path from $i$ to a node $t \in V$ with $e_t < 0$, then $\mu_i < \infty$. The following claim is also easy to verify.

**Claim 6.5.** $\Gamma_{ij}^{\mu+}(\alpha) = \frac{\mu_i}{\mu_j}\Gamma_{ij}^{+}(\alpha)$, *and* $\Gamma_{ij}^{\mu-}(\alpha) = \frac{\mu_i}{\mu_j}\Gamma_{ij}^{-}(\alpha)$. □

This claim implies that for an $s-t$ walk $P$, $\Gamma_f^{\mu+}(P) = \frac{\mu_s}{\mu_t}\Gamma_f^{+}(P)$, and thus for a cycle $C$, $\Gamma_f^{\mu+}(C) = \Gamma_f^{+}(C)$.

**Theorem 6.6** ([33])**.** *Let $f \in \mathbb{R}^E$ satisfy $0 \le f \le \tilde{u}$. Then the following are equivalent.*

*(i) $f$ is an optimal solution to the symmetric version.*

*(ii) $E_f$ contains no generalized augmenting paths.*

*(iii) There exists a conservative labeling $\mu$ with $e_i = 0$ whenever $1/M_i < \mu_i < \infty$.*

*Proof.* The equivalence of (i) and (iii) follows by the Karush-Kuhn-Tucker conditions, with $\mu_i$ being the reciprocal of the Lagrange multiplier corresponding to $e_i + \kappa_i \ge 0$. (i) implies (ii) by Lemma 6.4. The proof of (ii)⇒(iii) is the same as in Theorem 5.2, with $\gamma(P)$ replaced by $\Gamma_f^{+}(P)$. □

### 6.4 Δ-feasibility

In a Δ-phase of the algorithm, we will use a linearization to chunks of Δ as in [17].

$$\theta_\Delta^\mu(ij) := \begin{cases} \frac{\Gamma_{ij}^\mu(f_{ij}^\mu+\Delta)-\Gamma_{ij}^\mu(f_{ij}^\mu)}{\Delta} & \text{if } ij \in E,\\ \frac{\Delta}{\Gamma_{ji}^\mu(f_{ji}^\mu)-\Gamma_{ji}^\mu(f_{ji}^\mu-\Delta)} & \text{if } ji \in E, f_{ji}^\mu \ge \Delta,\\ \frac{f_{ji}^\mu}{\Gamma_{ji}^\mu(f_{ji}^\mu)-\Gamma_{ji}^\mu(0)} & \text{if } ji \in E, 0 < f_{ji}^\mu < \Delta \end{cases} \tag{6}$$

Relaxing the notion of conservativity, we say that for the relabeling $\mu$ is *Δ-arc-conservative*, if $\theta_\Delta^\mu(ij) \le 1$ for all $ij \in E_f$. (The reason for the term arc-conservative is that the condition $\mu_i = 1/M_i$ for $e_i < 0$ in the definition of conservativity is not imposed here. We will define Δ-conservativity later, after giving the appropriate notion of negative nodes.) The main use of this notion is due to the fact that Δ-arc-conservativity is maintained when sending Δ units of flow on arcs with $\theta_\Delta^\mu(ij) = 1$. This is formulated in the next simple lemma.

**Lemma 6.7.** *Assume $\mu$ is Δ-arc-conservative, and let $ij \in E_f$ be an arc with $\theta_\Delta^\mu(ij) = 1$. Δ-arc-conservativity is maintained if we increase $f_{ij}^\mu$ by Δ if $ij$ is a forward arc or a backward arc with $f_{ji}^\mu \ge \Delta$. If $ij$ is a backward arc with $f_{ji}^\mu < \Delta$, Δ-conservativity is maintained if decreasing $f_{ji}^\mu$ to 0.* □

By (5), if $\mu$ is a Δ-arc-conservative labeling then it is Δ′-arc-conservative for all $\Delta' \ge \Delta$. While the converse is not true, we can prove the following. (Recall that $d_i$ is the number of arcs incident to $i$, both incoming and outgoing.)

**Lemma 6.8.** *Assume $\mu$ is $2\Delta$-arc-conservative. We can modify each $f^\mu_{ij}$ by at most $\Delta$ so that $\mu$ becomes $\Delta$-arc-conservative. In the transformation, each $ij \in E$ accounts for an increase of at most $2\Delta$ and a decrease of at most $\Delta$ in the values $e^\mu_i$ and $e^\mu_j$. Consequently, each $e^\mu_i$ may increase by at most $2d_i\Delta$ or decrease by at most $d_i\Delta$.*

*Proof.* Consider an arc $ij \in E$. If $\theta^\mu_\Delta(ij) \le 1$ and $\theta^\mu_\Delta(ji) \le 1$, there is no need to modify $f^\mu_{ij}$.

Assume $\theta^\mu_\Delta(ij) > 1$. We claim that increasing $f^\mu_{ij}$ by $\Delta$ satisfies both $\theta^\mu_\Delta(ij) \le 1$ and $\theta^\mu_\Delta(ji) \le 1$. The inequality

$$\Gamma^\mu_{ij}(f^\mu_{ij} + \Delta) - \Gamma^\mu_{ij}(f^\mu_{ij}) > \Delta. \tag{7}$$

immediately gives $\theta^\mu_\Delta(ji) < 1$ for the new value. Since $\mu$ is $2\Delta$-arc-conservative, we have

$$\Gamma^\mu_{ij}(f^\mu_{ij} + 2\Delta) - \Gamma^\mu_{ij}(f^\mu_{ij}) \le 2\Delta. \tag{8}$$

Subtracting (7) from (8) gives $\theta^\mu_\Delta(ij) < 1$ after increasing.

Next, assume $\theta^\mu_\Delta(ji) > 1$ for the original value. (It can be shown that $\theta^\mu_\Delta(ji) > 1$ and $\theta^\mu_\Delta(ij) > 1$ may not occur simultaneously). If $f^\mu_{ij} \ge \Delta$, then let us decrease $f^\mu_{ij}$ by $\Delta$; otherwise, decrease it to 0. In the first case, we have

$$\Gamma^\mu_{ij}(f^\mu_{ij}) - \Gamma^\mu_{ij}(f^\mu_{ij} - \Delta) < \Delta, \tag{9}$$

immediately giving $\theta^\mu_\Delta(ij) < 1$ after decreasing. If $f^\mu_{ij} < \Delta$, then

$$\Gamma^\mu_{ij}(f^\mu_{ij}) - \Gamma^\mu_{ij}(0) < f^\mu_{ij} \tag{10}$$

By (5a), this implies

$$\Gamma^\mu_{ij}(\Delta) - \Gamma^\mu_{ij}(0) < \Delta,$$

which is equivalent to $\theta^\mu_\Delta(ij) < 1$ after decreasing.

If $f^\mu_{ij} < \Delta$, then after decreasing, $ji \notin E_f$, hence $\theta^\mu_\Delta(ji)$ is not defined. If $f^\mu_{ij} \ge 2\Delta$, then by $2\Delta$-conservativity we have

$$\Gamma^\mu_{ij}(f^\mu_{ij}) - \Gamma^\mu_{ij}(f^\mu_{ij} - 2\Delta) \ge 2\Delta. \tag{11}$$

Subtracting (9) from this gives $\theta^\mu_\Delta(ji) < 1$. Finally, if $\Delta \le f^\mu_{ij} \le 2\Delta$, the $2\Delta$-conservativity yields

$$\Gamma^\mu_{ij}(f^\mu_{ij}) - \Gamma^\mu_{ij}(0) \ge f^\mu_{ij}. \tag{12}$$

Subtracting again (9) from this gives $\theta^\mu_\Delta(ji) < 1$.

Let us now turn to the second part of the proof. If we increase $f^\mu_{ij}$ by $\Delta$, then $e^\mu_i$ decreases by $\Delta$ and $e^\mu_j$ increases by at most $2\Delta$, because of (8) and monotonicity. If we decrease $f^\mu_{ij}$, then $e^\mu_i$ increases by at most $\Delta$, and $e^\mu_j$ decreases by at most $\Delta$, using (9) or (10). Hence the total change of any $e^\mu_i$ is between $-d_i\Delta$ and $2d_i\Delta$. $\square$

The subroutine Adjust($\Delta$) performs the modifications described in the proof at the beginning of all but the first $\Delta$-phase.

Adjust($\Delta$) could transform a node with $e_i \ge 0$ to $e_i < 0$. For this reason, we keep a reserve excess $d_i\Delta$ in node $i$ as follows. As for generalized flows, we introduce arc imbalances, denoted by $\bar{e}_{ij}$ for every arc $ij \in E$. The modified node excess $\bar{e}_i$ is defined by

$$\bar{e}_i = e_i - \sum_{ji \in E} \bar{e}_{ji} - d_i\Delta\mu_i \quad \forall i \in V \tag{13}$$

We define $\bar{e}_i^\mu = \bar{e}_i/\mu_i$ and $\bar{e}_{ij}^\mu = \bar{e}_{ij}/\mu_j$ as in the generalized flow setting, and also maintain $0 \le \bar{e}_{ij}^\mu < \Delta$. However, if $\mu_i = \infty$, (13) suggests to define $\bar{e}_i^\mu = -d_i\Delta$.

We call node $i$ is *positive* if $\bar{e}_i^\mu > 0$, *negative* if $\bar{e}_i^\mu < 0$ and *neutral* $\bar{e}_i^\mu = 0$. Let $N$ denote the set of negative nodes. Now we are ready to define $\Delta$-conservativity: $\mu$ is $\Delta$-*conservative*, if $\Delta$-arc-conservative, and $\mu_i \ge 1/M_i$ for every $i \in V$ with equality whenever $i \in N$.

In the $\Delta$-phase, after ADJUST($\Delta$), we will maintain a $\Delta$-conservativeness and also $\bar{e}_{ji} = 0$ whenever $i \in N$, $ji \in E$. Note that $e_i < 0$ implies $\bar{e}_i < 0$. If we achieve $N = \emptyset$ at some point of the algorithm, then $e_i \ge \bar{e}_i \ge 0$ for all $i \in V$ and thus we have an optimal solution. Furthermore, unlike for generalized flows, no node $i$ may have $\mu_i = \infty$ in a $\Delta$-conservative labeling. This is because it would imply $i \in N$ since $\bar{e}_i^\mu = -d_i\Delta$, although $\mu_i = 1/M_i$ for all negative nodes.

When decreasing the scaling factor from $2\Delta$ to $\Delta$, we set all arc imbalances to 0 and perform ADJUST($\Delta$). Before ADJUST($\Delta$), each $\bar{e}_i$ value increases by at least $d_i\Delta\mu_i$ due to change in the scaling factor (as a consequence, all previously positive or neutral nodes will have $\bar{e}_i^\mu \ge d_i\Delta$). By Lemma 6.8, we get the following:

**Lemma 6.9.** *If we have a $2\Delta$-conservative labeling at the end of the $2\Delta$-phase, then applying* ADJUST($\Delta$) *at the beginning of a $\Delta$-phase gives a $\Delta$-conservative labeling.* □

While $f_{ij} > \tilde{u}_{ij}$ is possible, in a $\Delta$-conservative labeling $f_{ij}^\mu < \tilde{u}_{ij}^\mu + \Delta$. This is since otherwise $\Gamma_{ij}^\mu(f_{ij}^\mu) = \Gamma_{ij}^\mu(f_{ij}^\mu - \Delta)$, giving $\theta_\Delta^\mu(ji) = \infty$.

## 6.5 $\Delta$-canonical labelings

Given a pseudoflow $f$ and a $\Delta$-conservative labeling $\mu$, the arc $ij \in E_f$ is called tight if $\theta_\Delta^\mu(ij) = 1$. A directed path in $E_f$ is called tight if it consists of tight arcs. $\mu$ is a $\Delta$-*canonical* labeling, if from each node $i$ there exists a tight path to a negative or to a neutral node.

This is weaker than the notion of canonical labelings for generalized flows, where we wanted a tight path from each positive or neutral node to a negative one. On the other hand, this is required for every node (as remarked above, there are no nodes $i$ with $\mu_i = \infty$). The reason why the algorithm in Section 5.2 cannot be directly applied, is that increasing $\mu_i$ could turn a nonnegative node to negative, and thus we would possibly get negative nodes $i$ with $\mu_i > 1/M_i$. We present a modification of that algorithm, called $\Delta$-TIGHTEN-LABEL$(f, \mu)$, that returns a $\Delta$-canonical label $\mu' \ge \mu$ for a $\Delta$-conservative label $\mu$.

In any phase of the algorithm, we maintain a subset $S \subseteq V$ of nodes $i$ for which there exists a (possibly empty) tight path for $\mu$ to a neutral or a negative node. $S$ is initialized as the set of negative and neutral nodes, and is extended by at least one element per phase. The algorith terminates when $S = V$.

Let $ij \in E_f$ be an arc with $i \in V - S$, $j \in S$. By the definition of $S$, $\theta_\Delta^\mu(ij) < 1$. Increasing $\mu_i$ and not changing $\mu_j$, the value of $\theta_\Delta^\mu(ij)$ increases. This can be seen by rewriting $\theta_\Delta^\mu(ij)$ in terms of the original $\Gamma_{ij}$ and $f_{ij}$. For example, if $ij \in E$, then

$$\theta_\Delta^\mu(ij) = \frac{\Gamma_{ij}(f_{ij} + \mu_i\Delta) - \Gamma_{ij}(f_{ij})}{\mu_j\Delta},$$

and if $ji \in E_f$, $f_{ji}^\mu \ge \Delta$, then

$$\theta_\Delta^\mu(ij) = \frac{\mu_i\Delta}{\Gamma_{ji}(f_{ji}) - \Gamma_{ji}(f_{ji} - \mu_j\Delta)}.$$

Let $\alpha_{ij}$ denote the value of $\alpha$ so that multiplying $\mu_i$ by $\alpha$ gives $\theta^\mu_\Delta(ij) = 1$; let $\alpha_{ij} = \infty$ if no such value exists. If $ij$ is a backward arc, $\alpha_{ij} = 1/\theta^\mu_\Delta(ij)$, and if $ij$ is a forward arc, then it can be obtained using an inverse oracle call.

For a node $i \in V' - S$, $\bar{e}^\mu_i$ is defined by (13), hence $\mu_i$ can be multiplied by at most $\alpha_i = (e^\mu_i - \sum_{ji\in E} \bar{e}_{ji})/(d_i\Delta\mu_i)$. Let us define

$$\alpha = \min\left\{\min\{\alpha_{ij} : ij \in E_f, i \in V - S, j \in S\}, \min\{\alpha_i : i \in V - S\}\right\}.$$

Let us multiply all $\mu_i$ by $\alpha$ for all $i \in V - S$. By the definition of $S$, $\alpha > 1$, and after multiplying by $\alpha$, either at least one arc $ij \in E_f$ with $i \in V - S$, $j \in S$ becomes tight, or a new neutral node is created. $S$ is thus extended by at least one new node.

To verify that $\mu$ remains $\Delta$-conservative, we also have to check $\theta^\mu_\Delta(ij) \le 1$ on all arcs $ij \in E_f$, $i, j \in V - S$. This follows by (5).

## 6.6 The main algorithm

Let us initialize $\mu_i = 1/M_i$ for every $i \in V$, $f_{ij} = 0$ if $\Gamma_{ij}(0) > -\infty$ and $f_{ij} = \tilde{u}_{ij}$ if $\Gamma_{ij}(0) = -\infty$, and set all arc imbalances $\bar{e}_{ij} = 0$. Let us pick the initial value $\Delta = MU$.

Let $Ex^\mu_\Delta(f) = \sum_{i\in V-N} \bar{e}^\mu_i$ denote the total modified node excess. Each $\Delta$ phase consists of a preprocessing part and a main part. After the preprocessing part, we will have a $\Delta$-conservative labeling with $Ex^\mu_\Delta(f) \le (2n+8m)\Delta$. At the end of the main part, we will have $\bar{e}^\mu_i \le \Delta$ for each $i \in V$.

Preprocessing in the first phase consists of giving an initial solution as described above. In all later phases, we have a pseudoflow $f$ with a $2\Delta$-conservative labeling from the previous phase. We set all arc imbalances to zero, and perform Adjust($\Delta$).

Now we turn to the main part of the $\Delta$-phase. Let $D$ denote the set of nodes $i$ with $\bar{e}^\mu_i > \Delta$ (unlike for generalized flows, where we had $\bar{e}^\mu_i \ge \Delta$). The main part consists of iterations, and terminates whenever $D$ becomes empty.

We start every iteration by updating $\mu$ to a canonical labeling by calling $\Delta$-Tighten-Label($f, \mu$). If $D \ne \emptyset$ still holds, pick an arbitrary $s \in D$, and identify a tight path $P$ to a neutral or negative node $t$. Apply Concave-Push($P$), and move to the next iteration.

Concave-Push($P$) is almost identical to Push($P$), with the only difference is that on $P = i_0i_1\dots i_k$ with $i_0 = s$ and $i_k = t$, we stop pushing flow whenever $\bar{e}^\mu_{i_h} \le \Delta$ (instead of strict inequality).

## 6.7 Analysis

**Claim 6.10.** *The inital $\mu$ is $\Delta$-conservative.*

*Proof.* $\mu_i\Delta \ge U$ is straightforward. If $f_{ij} = 0$, then $ji \notin E_f$, and $\theta^\mu_\Delta(ij) \le 1$, since by definition, $\Gamma_{ij}(\tilde{u}_{ij}) - \Gamma_{ij}(0) \le U$. If $f_{ij} = \tilde{u}_{ij}$, then $\theta^\mu_\Delta(ij) = 0$ since $\Gamma_{ij}(\tilde{u}_{ij}) = \Gamma_{ij}(\tilde{u}_{ij} + \mu_i\Delta)$. $ji \in E_f$, but $\theta^\mu_\Delta(ji) = 0$, since the denominator is $\infty$. □

$\Delta$-conservativity is maintained in the main part of the $\Delta$-phase by Lemma 6.7.

**Claim 6.11.** *At the end of the preprocessing part, we have a $\Delta$-conservative labeling with $Ex^\mu_\Delta(f) \le (2n + 8m)\Delta$.*

*Proof.* For the inital solution in the first phase, either $f_{ij} = \Gamma_{ij}(f_{ij}) = 0$ or $f_{ij} = \tilde{u}_{ij}$. Hence $Ex^\mu_\Delta(f) \le M(\sum_{i\in V} |b_i| + mU) \le (m + n)MU$. The claim follows, since $\Delta = MU$.

In a later phase, we have a $2\Delta$-conservative labeling from the end of the previous $2\Delta$-phase with $\bar{e}_i^\mu \le 2\Delta$ and $\bar{e}_{ij}^\mu \le 2\Delta$. When setting the scaling factor to $\Delta$, each $\bar{e}_i^\mu$ increases by $d_i\Delta$ due to the change in (13) when replacing $2\Delta$ by $\Delta$.

Setting the arc imbalances to 0 increases $Ex_\Delta^\mu(f)$ by at most $2m\Delta$. So far, $Ex_\Delta^\mu(f) \le (2n+4m)\Delta$. By Lemmas 6.8 and 6.9, Adjust($\Delta$) yields a $\Delta$-conservative relabeling, increasing each $\bar{e}_i^\mu$ by at most $2d_i\Delta$. Hence at the end of the preprocessing phase, $Ex_\Delta^\mu(f) \le (2n+8m)\Delta$. □

The next claim can be proved identically as Claim 5.5.

**Claim 6.12.** *During the entire $\Delta$-phase, (13) is maintained and $e_{ij}^\mu < \Delta$ for all arcs $ij \in E$.* □

Next, we prove the analogue of Lemma 5.7.

**Lemma 6.13.** *A $\Delta$-phase terminates with at most $2n+8m$ flow augmentations.*

*Proof.* For each $i \in V-N$, let us define $\psi_i = \lfloor \bar{e}_i^\mu/\Delta \rfloor$ if $e_i^\mu > \Delta$ and $\psi_i = 0$ otherwise. Let $\Psi = \sum_{i\in V-N} \psi_i$. (The only difference to the proof of Lemma 5.7 is that $\Psi_i = 0$ if $e_i^\mu = \Delta$). By Claim 6.11, the initial value is at most $2n+8m$. We claim that $\Psi$ decreases by at least one at each call of Concave-Push($P$). The proof is the same as that of Lemma 5.7, with the only difference that $\psi_{i_{h+1}}$ decreases by 1 whenever $\bar{e}_{i_{h+1}}^\mu$ becomes $\Delta$ or less. This is always be the case when Concave-Push($P$) terminates. □

Recall that $\kappa_f = \sum_{i\in V} M_i\kappa_i = \sum_{i\in V} M_i \min\{-e_i, 0\}$ denotes the excess discrepancy. Since all nodes $i$ with $e_i < 0$ are contained in $N$, $e_i^\mu = M_i\kappa_i$ for any $\Delta$-conservative labeling $\mu$. The next theorem shows that if $\Delta < \varepsilon/(2n+8m)$, then we have a $\varepsilon$-optimal solution at the end of the $\Delta$-phase.

**Theorem 6.14.** *At the end of phase $\Delta$, the actual $f$ is $(2n+8m)\Delta$-optimal.*

*Proof.* Let us keep running the algorithm forever unless it finds a 0-discrepancy solution at some phase. First, consider the case when for some $\Delta' = \Delta/2^k$, we terminate with a 0-discrepancy solution. In all phases between $\Delta$ and $\Delta'$ , the total decrease of excess discrepancy is bounded by $(2n+8m)(\Delta/2 + \Delta/4 + \ldots + \Delta/2^k) \le (2n+8m)\Delta$. Since in the $\Delta'$-phase we have a 0-discrepancy solution, the total discrepancy at the end of the $\Delta$-phase is at most $(2n+8m)\Delta$, proving the theorem.

Assume now the procedure runs forever. For each $i \in V$, $\kappa_i$ is decreasing and thus converges to some limit $\kappa_i^*$. Let $\kappa^* = \sum_{i\in V} M_i\kappa_i^*$. As above, the total decrease of the excess discrepancy after phase $\Delta$ is bounded by $(2n+8m)\Delta$, hence $\kappa_f \le \kappa^* + (2n+8m)\Delta$. The proof finishes by constructing an optimal pseudoflow $f^*$ with discrepancy $\kappa^*$.

Let $f^{(t)}$ denote the flow at time $t$, for $\Delta^{(t)} = \Delta_0/2^t$, with labels $\mu_i^{(t)}$. For each node $i$, $\mu_i^{(t)}$ is increasing; let $\mu_i^* = \lim_{t\to\infty} \mu_i^{(t)}$. Let $V_\infty = \{i : \mu_i^* = \infty\}$. We claim that $V - V_\infty \ne \emptyset$. Indeed, if $i \in N$, then $\mu_i = 1/M_i$, and if $i \notin N$ in a certain phase, it would not enter $N$ again. Consequently, the set of negative nodes is decreasing. If it gets empty, then we arrive at a 0-discrepancy solution. If not, then we have a set $N^*$ which remains the set of negative nodes after a finite number of steps and thus $\mu_i^* = 1/M_i$ for $i \in N^*$.

We also claim that for each $i$, the sequence $\Delta^{(t)}\mu_i^{(t)}$ is bounded. Indeed, if $\mu_i^* < \infty$, it converges to 0. If $\mu_i^* = \infty$, then after a finite number of steps, $i \notin N$, thus $d_i\Delta^{(t)}\mu_i^{(t)} \le e_i \le \sum_{ji\in E} \Gamma_{ij}(\tilde{u}_{ij}) - b_i$. Then $f_{ij}^{(t)}$ is also a bounded sequence, since $0 \le f_{ij}^{(t)} \le \tilde{u}_{ij} + \Delta^{(t)}\mu_i^{(t)}$ is implied as otherwise $\theta_{\Delta^{(t)}}^{\mu^{(t)}}(ji) = \infty$.

Consequently, we can choose an infinite set $T' \subseteq \mathbb{N}$ so that restricted to $t \in T'$, all sequences $f_{ij}^{(t)}$ and all sequences $\Delta^{(t)}\mu_i^{(t)}$ converge. Let $f_{ij}^*$ and $D_i^*$ denote the limits, respecitvely ($D_i^* = 0$ if $\mu_i^* < \infty$).

Observe that $f^*$ has excess discrepancy $\kappa^*$. We shall prove that $f^*$ is an optimal pseudoflow with optimal labeling $\mu_i^*$, completing the proof.

Let $e_i^*$ denote the excesses. If $e_i^* < 0$, then clearly $i \in N^*$ and $\mu_i^* = 1/M_i$. If $e_i^* > 0$, we shall prove $\mu_i^* = \infty$. For a contradiction, assume $\mu_i^* < \infty$. Then for sufficiently large $t \in T'$, $(d_i + 2n + 8m)\Delta^{(t)}\mu_i^{(t)} < e_i^{(t)}$ and thus $Ex^{\mu}_{\Delta^{(t)}}(f) > (2n + 8m)\Delta^{(t)}$, a contradiction.

We have to prove $\Gamma_{ij}^{\mu^*+}(f_{ij}^{*\mu^*}) \le 1$ whenever $ij \in E_{f^*}$. This is easy to see if $\mu_i^*, \mu_j^* < \infty$: if $ij$ is a forward arc, then

$$1 \ge \theta^{\mu^{(t)}}_{\Delta^{(t)}}(ij) = \frac{\Gamma_{ij}(f_{ij}^{(t)} + \Delta^{(t)}\mu_i^{(t)}) - \Gamma_{ij}(f_{ij}^{(t)})}{\Delta^{(t)}\mu_i^{(t)}} \cdot \frac{\mu_i^{(t)}}{\mu_j^{(t)}}. \tag{14}$$

The first fraction converges to $\Gamma_{ij}^{+}(f_{ij}^*)$, while the second to $\mu_i^*/\mu_j^*$, leading to the conclusion using Claim 6.5. The proof is analogous if $ij$ is a backward arc, using that for sufficiently large $t \in T'$, $\Delta^{(t)}\mu_j^{(t)} < f_{ji}^{(t)}$, and thus

$$1 \ge \theta^{\mu^{(t)}}_{\Delta^{(t)}}(ij) = \frac{\Delta^{(t)}\mu_j^{(t)}}{\Gamma_{ji}(f_{ji}^{(t)}) - \Gamma_{ji}(f_{ji}^{(t)} - \Delta^{(t)}\mu_j^{(t)})} \cdot \frac{\mu_i^{(t)}}{\mu_j^{(t)}}. \tag{15}$$

By the convention on infinite $\mu$ values, if $\mu_j^* = \infty$ then $\Gamma_{ij}^{+}(f_{ij}^*) = 0$. It is left to prove that if $ij \in E_{f^*}$, $\mu_i^* = \infty$, $\mu_j^* < \infty$, then $ij$ is a forward arc with $f_{ij}^* \ge \tilde{u}_{ij}$. (If $f_{ij}^* > \tilde{u}_{ij}$, we may decrease $f_{ij}^*$ to $\tilde{u}_{ij}$, thereby only increasing $e_i^*$.)

Assume first $ij$ is a forward arc. If $D_i^* = 0$, then the first term in (14) converges to $\Gamma_{ij}^{+}(f_{ij}^*)$. If $\Gamma_{ij}^{+}(f_{ij}^*) > 0$ then the right hand side converges to $\infty$, a contradiction. Hence $\Gamma_{ij}^{+}(f_{ij}^*) = 0$, yielding $f_{ij}^* \ge \tilde{u}_{ij}$. If $D_i^* > 0$ and $f_{ij}^* < \tilde{u}_{ij}$, then both the numerator and denominator of the first fraction converge to positive numbers, a contradiction again.

Finally, assume $ij$ is a backward arc, that is, $f_{ji}^* > 0$. (15) holds for sufficiently large $t \in T'$. The first fraction converges to $1/\Gamma_{ji}^{-}(f_{ji}^*)$, while the second to $\infty$, a contradiction. $\square$

**Theorem 6.15.** *The above algorithm finds a $\varepsilon$-approximate solution to the symmetric concave generalized flow problem in $O(m(m + n \log n) \log(MUm/\varepsilon))$ oracle calls.*

*Proof.* The initial value of $\Delta$ is $MU$, and we terminate if $\Delta < \varepsilon/(2n + 8m)$ by Theorem 6.14. Hence the total number of scaling phases is $O(\log(MUm/\varepsilon))$. The number of iterations in a phase is $O(m)$ by Lemma 6.13, and the running time of an iteration is dominated by $\Delta$-Tighten-Label, a slightly modified version of Dijkstra's algorithm that can be implemented in $O(m+n \log n)$ time using Fibonacci heaps as in [10] $\square$

# 7 Sink versions of the problems

In this section, we show how the algorithms in Sections 5 and 6 can be applied to solve the to the sink versions of the corresponding problems.

For generalized flows, let us set $M_t = 1$ and $M_i = B^n + 1$ for every $i \in V - t$. Let us set $b_t = \sum_{j:jt\in E} \gamma_{jt}u_{jt} - \sum_{j:tj\in E} \ell_{tj} + 1 \le d_t B^2 + 1$. This a strict upper bound on $\sum_{j:jt\in E} \gamma_{jt} f_{jt} - \sum_{j:tj\in E} f_{tj}$, hence $e_t < 0$ will hold for any pseudoflow.

Let us run the algorithm for the symmetric formulation with these $M_i$'s, returning an optimal pseudoflow $f$ and optimal labels $\mu$. We claim that $F$ is also optimal for the sink formulation. If $e_i \ge 0$ for all $i \ne t$, this is clearly the case.

On the other hand, if $e_i < 0$ for some nodes $i \neq t$, then no feasible solution exists. To verify this, we show that no path in $E_f$ may exist from $t$ to such a node. Indeed, on each path $P$, $\gamma(P) \leq B^n$. Hence the existence of such a path would yield $1 = \mu_t \leq \gamma(P)\mu_i < 1$, a contradiction.

Let $V' \subseteq V$ be the set of nodes $j$ with a path in $E_f$ from $j$ to some $i$ with $e_i < 0$. Let us set $\mu'_i = \mu_i$ if $i \in V'$ and $\mu'_i = \infty$ otherwise. Now $\mu'_t = \infty$, hence $\mu'_t$ is an optimal labeling for the modified problem when we change $b_t$ to an arbitrary negative value. This shows that no feasible solution may exist.

By setting the $b_t$ value and the $M_i$'s, $B$ has increased to $d_t B^2 + 1$ and $M = B^n + 1$. This gives running time $O(m^3 \log m \log(d_t B))$.

Let us turn to concave generalized flows. An $\varepsilon$-approximate solution to the sink version means a pseudoflow $f$ with $\sum_{i \in V-t} \kappa_i \leq \varepsilon$ and $e_t$ being at least the optimum value minus $\varepsilon$.

Let us set $b_t = U^* + 1$, a strict upper bound on $\sum_{j:jt \in E} \gamma_{jt} f_{jt} - \sum_{j:tj \in E} f_{tj}$ ($U^*$ was defined in Section 2.1). Thus $e_t < 0$ is always guaranteed. $U^*$ is also a lower bound on to the pseudoflow in $t$; hence $|\kappa_t| \leq b_t + U^* \leq 2U^* + 1 = D$.

Let us set $M_i = D/\varepsilon$ if $i \in V - t$ and $M_t = 1$. Let us run the algorithm for the symmetric formulation to obtain a $\varepsilon$-optimal solution $f$.

If $\kappa_f > 2D$, then no solution with $\sum_{i \in V-t} \kappa_i \leq \varepsilon$ may exist. Indeed, if there existed such a solution, then $\kappa_f \leq D + (D/\varepsilon)\varepsilon = 2D$ since $|\kappa_t| \leq D$ .

If $\kappa_f < 2D$, we can similarly conclude $\sum_{i \in V-t} \kappa_i < \varepsilon$. Also $\kappa_t$ cannot be further than $\varepsilon$ from the optimum because of $M_t = 1$, and since the optimal value of $e_t$ in this problem is at least that for the sink version.

This gives a running time bound $O(m(m + n \log n) \log(U^* m/\varepsilon))$.

## 8 Finding the optimal solution for rational convex programs

In this section, we first give a general theorem which shows how an approximate solution to the sink version can be converted to an exact optimal solution, given that one exists. We shall verify the required technical properties with appropriate parameters for the linear Fisher market model. They should also hold for the extensions discussed in Section 3 as well, giving polynomial time algorithms for finding optimal solutions. Unlike the linear Fisher model, ADNB might be infeasible. However, it can be shown that if the problem is infeasible, then for appropriate (polynomially small) $\varepsilon$, the $\varepsilon$-approximate version is also infeasible.

**Theorem 8.1.** *Let problem $\mathcal{P}$ be given by the sink formulation with $n$ nodes and $m$ arcs, and complexity parameters $U$, $U^*$. Assume $\mathcal{P}$ is guaranteed to have a rational optimal solution, and the following conditions hold for some values $\varepsilon, T$ and $\tau$.*

*(P1) Consider the algorithm for the sink version for an $\varepsilon$-approximation. Then $\mu_i \leq T$ holds for any $i \in V$, even if running the algorithm for an arbitrary number of phases.*

*(P2) Assume that for each $ij \in E$, we are given an interval $I_{ij} \subseteq [\ell_{ij}, u_{ij}]$ with $|I_{ij}| \leq 2T\varepsilon$, with the guarantee that there exists an optimal solution $f^*$ with $f^*_{ij} \in I_{ij}$ for all $ij \in E$. Then there exists an algorithm that finds an optimal solution $\tilde{f}$ in running time $\tau$ ($\tilde{f} = f^*$ is not required).*

*Then there exist an algorithm for finding the exact optimal with $O(m(m + n \log n) \log(U^* m/\varepsilon)) + \tau$ running time.*

*Proof.* Assume we run the algorithm for the sink version for $\varepsilon$-approximation forever, as in the proof of Theorem 6.14. The $\mu_j$'s shall converge to some finite values $\mu_j^* \leq T$. In any $\Delta$-phase, the total change of $f_{ij}^{\mu}$ is bounded by $\varepsilon' = (2n+8m)\Delta$, and thus $f_{ji}$ may change by at most $T\varepsilon'$. Therefore all $f_{ij}$'s converge to some values $f_{ij}^*$, which can be seen to give an optimal solution, as in the proof of Theorem 6.14. (Indeed, the proof simplifies, since the main difficulties were because of the possibly infinite $\mu_i$ values).

The algorithm terminates whenever $\Delta < \varepsilon/(2n+8m)$. At this point, the intervals $I_{ij} = [f_{ij} - T\varepsilon, f_{ij} + T\varepsilon]$ satisfy the conditions in (P1), since $|f_{ij} - f_{ij}^*| \leq T\varepsilon$. Running the $\varepsilon$-approximation algorithm and then the algorithm in (P1) gives the running time bound. $\square$

To ensure property (P1), a useful method is to enforce the existence of a unique optimal solution by perturbing the input data, as done by Orlin [27] for linear Fisher markets. If there is a unique rational optimal solution $f^*$ with all entries having denominator at most $Q$, then setting $2T\varepsilon < 1/Q$ enables us to identify the set of arcs with $f_{ij}^* > 0$. This can be already enough to compute $f^*$ efficiently.

## 8.1 Application for linear Fisher markets

Let us now apply Theorem 8.1 for linear Fisher markets. Let us assume all utilities $U_{ij}$ and budgets $m_i$ are nonnegative integers, with $U_{\max} = \max\{U_{ij} : i \in B, j \in G\}$, $R = \sum_{i \in B} m_i$. Let $n = |G| + |B|$ and let $m$ be the number of pairs $ij$ with $U_{ij} > 0$; in the concave generalized flow instance, the number of nodes is $n+1$ and the number of arcs is $m + |B|$. Let us assume that there exists at least one arc with positive utility with incident to any buyer and to any good.

Consider a candidate solution to the linear Fisher market, with price $p_j$ for each good $j \in G$. Let $x_{ij} \geq 0$ denote the amount of good $j$ purchased by buyer $i$.

$(p, x)$ is an optimal solution if and only if (i) $\sum_{i \in B} x_{ij} = 1$ for each good $j$, that is, each good is fully sold; and (ii) for any buyer $i$ and good $j$, $U_{ij}/p_j \leq \sum_{j \in G} U_{ij} x_{ij}/m_i$, and equality holds whenever $x_{ij} > 0$.

An equivalent characterization can be given by the money allocation variables, $y_{ij} = p_j x_{ij}$. $(p, y)$ is optimal if $\sum_{i \in B} y_{ij} = p_j$, $\sum_{i \in B} y_{ij} = m_i$, and as in (ii), for any buyer $i$, all goods with $y_{ij} > 0$ maximize $U_{ij}/p_j$.

By strict concavity of the objective, the utilities $\sum_{j \in G} U_{ij} x_{ij}$ accrued by the players are the same in any optimal solution. Yet these same values can be obtained by different allocations. As in [27], we assume that there exist a unique optimal allocation as well. This can be done by a lexicographic perturbation of the $U_{ij}$ values, without significantly increasing the running time. This guarantees that the set of arcs with $x_{ij} > 0$ in an optimal solution is cycle-free.

For the concave generalized flow instance, let us set upper capacity $v_{ji} = 2$ on each arc $ji$ with $j \in G$, $i \in B$, and $v_{it} = 2\sum_{j \in G} U_{ij}$ on each arc $it$ (we set larger capacities so that the capacity constraints would never become tight). Hence the complexity parameter $U$ is bounded by $2|G|U_{\max}$.

We also need to give a bound $U^*$ as in Section 2.1. Let us define $f_{ji} = 1/d_j$ for all arcs $ji$, and $f_{it} = \sum_{ji \in E} U_{ij} f_{ji}$ for each $i \in B$. This gives $e_t \geq \sum_{i \in B} m_i \log(1/n)$. Hence $U^* = \max\{R \log n, |B|U\} \leq \max\{R \log n, 2n^2 U_{\max}\}$.

We shall show that (P1) and (P2) hold with $T = 3U^*$ and $\varepsilon = 1/(6nRU_{\max}^n U^*)$. To simplify the complexity bound, let us assume $R \log n \leq 2n^2 U_{\max}$. Then Theorem 8.1 gives a total running time $O(mn(m + n\log n)\log U_{\max})$.

(P2) follows by the next lemma, since giving an interval of length $2T\varepsilon$ containing each $x_{ij}^*$ provides the set of positive $x_{ij}^*$ values.

**Lemma 8.2.** *Assuming that there exists a unique optimal allocation $x^*$, all positive $x^*_{ij}$ values are at least $1/(nRU^n_{\max})$.*

*Proof.* $x^*$ and the optimal prices $p^*$ can be uniquely obtained given the set $F$ of arcs $ji$ with $x_{ij} > 0$. By the uniqueness assumption, each component of $F$ is a tree. We can determine the allocations and prices for different components independently.

Let us add the opposite of arcs in $F$, setting $U_{ij} = 1/U_{ji}$ on $ij$ if $ji \in F$. Fixing the price of a good $j_0$ as $p_{j_0} = \alpha$ in a component of $F$, it uniquely determines $p_j$ for any good $j$ in the same component as $\alpha$ times the product of the $U_{ab}$ values on the unique path from $j_0$ to $j$. Using that the sum of the prices in a component should equal the sum of the budgets, this uniquely determines all prices, and all of them will have a common denominator $S \le nU^n_{\max}$. Then the optimal money allocations, given as $p_j x_{ij}$, can be easily obtained using the tree structure of $F$ in the component, having the same common denominator $S$. Each price is at most $R$, hence each positive $x_{ij}$ value is at least $1/(SR)$, proving the claim. $\square$

The proof also suggest an efficient, linear time algorithm for finding an equilibrium allocation. Therefore $\tau$ is neglibile compared to the running time of the approximation algorithm.

It is left to verify (P1). We will need the following simple bound.

**Claim 8.3.** *If $|\alpha| \le 1$ then $e^\alpha \le 1 + \alpha + 2\alpha^2$.* $\square$

**Lemma 8.4.** *$\mu_k \le 3U^*$ holds for any $k \in B \cup G$ in arbitrary $\Delta$-phase.*

*Proof.* Since the $\mu_k$ values are monotonely increasing, we may assume that $\Delta$ is sufficiently small: let $\Delta < 1/(6U^*)$. Recall from Section 7 that $M_t = 1$ and $e_t < 0$, hence $\mu_t = 1$.

Let us consider the inequality $\theta^\mu_\Delta(it) \le 1$. First, assume $f_{it} + \Delta\mu_i > v_{it}$ for the upper capacity $v_{it}$ on this arc. Then $\theta^\mu_\Delta(it) \le 1$ gives $\log(v_{it}) - \log(f_{it}) \le \Delta/m_i$. Since $\mu_i > 1/M_i$, we have $e_i \ge 0$ and thus $f_{it} \le \sum_{j\in G} U_{ji} f_{ji} \le (1+\varepsilon)\sum_{j\in G} U_{ji} < v_{it}/1.5$, giving $\log 1.5 \le \Delta/m_i$, a contradiction. ($f_{ji} \le 1+\varepsilon$ since $e_j \ge -\varepsilon$ by $\varepsilon$-optimality).

Therefore $f_{it} + \Delta\mu_i \le v_{it}$ and so $\theta^\mu_\Delta(it) \le 1$ gives $\log(f_{it} + \Delta\mu_i) - \log(f_{it}) \le \Delta/m_i$, or equivalently, $1 + \frac{\Delta\mu_i}{f_{it}} \le e^{\Delta/m_i}$. Using Claim 8.3, this yields $\mu_i \le f_{it}\left(\frac{1}{m_i} + \frac{2\Delta}{m_i^2}\right)$. Since $f_{it} \le v_{it}$, it follows that $\mu_i \le (1+\varepsilon)v_{it} \le 3U^*$.

Consider now a good $j \in G$. We first claim $\Delta\mu_j < 1$. Indeed, otherwise $\theta^\mu_\Delta(ji) \le 1$ would give $U_{ij}(2 - f_{ji}) \le \Delta\mu_i$, thus $f_{ji} \ge 2 - \frac{\Delta\mu_i}{U_{ij}}$, which is a contradiction to $\varepsilon$-optimality, using $\mu_i \le 3U^*$.

Therefore $\theta^\mu_\Delta(ji) \le 1$ gives $U_{ij}\mu_j \le \mu_i$ and thus $\mu_j \le 3U^*$ as well. $\square$

Finally, let us remark that the proof techniques of Lemma 8.4 can be extended to show the following structure of approximate solutions.

**Lemma 8.5.** *Assume $f$ is a $\varepsilon$-approximate solution for $\varepsilon \le 1/(6U^*)$. Then setting $y_{ij} = f^\mu_{ji}$, $p_j = 1/\mu_j$ provides a solution with $(1-\varepsilon)p_j \le \sum_{i\in B} y_{ij} \le p_j$, $(1-\varepsilon)m_j \le \sum_{j\in G} y_{ij} \le m_j$, and $y_{ij} > 0$ implies that $j$ maximizes $U_{ij}/p_j$.*

Also, notice that the algorithm runs in a fundamentally different way as [6] or [27]: while both these algorithms increase the prices, our algorithm works the other way around: it starts with the highest possible prices, and decreases them.

## 9 Discussion

We have given the first polynomial time combinatorial algorithms for both the symmetric and the sink formulation of the concave generalized flow problem.

As discussed in Section 4.4, it seems difficult to extend any generalized flow algorithm having separate cycle canceling and flow transportation subroutines. While this includes the majority of combinatorial algorithms, there are some exceptions. Goldberg, Plotkin and Tardos [12] gave two different algorithms: besides Fat-Paths, they also presented MCF, an algorithm that uses a minimum-cost circulation algorithm directly as a subroutine. Hence for the concave setting, it could be possible to develop a similar algorithm using a minimum concave cost circulation algorithm, for example [17] or [22] as a black box.

Another approach that avoids scaling is [39] for minimum-cost generalized flows and [31] for generalized flows: these algorithms can be seen as extensions of the cycle cancelling method, extending minimum mean cycles to GAP's in a certain sense. While it does not seem easy, it might be possible to develop such an algorithm for concave generalized flows as well.

In defining an $\varepsilon$-approximate solution for the sink version of concave generalized flows, we allow two types of errors, both for the objective and for feasibility. A natural question is if either of these could be avoided. While the value oracle model we use seems to need feasibility error, it might be possible to avoid it using a stronger oracle model as in [22]. Or we may also require a feasible solution as part of the input, as a starting point to maintain feasibility (For example if all lower bounds and node demands are 0 and $\Gamma_{ij}(0) = 0$ on all arcs $ij$, then $f \equiv 0$ is always feasible).

### Acknowledgments

The author is grateful to Vijay Vazirani for several fruitful discussions on market equilibrium problems.

## References

[1] D. Ahlfeld, J. Mulvey, R. Dembo, and S. Zenios. Nonlinear programming on generalized networks. *ACM Transactions on Mathematical Software (TOMS)*, 13(4):350–367, 1987.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc., Feb. 1993.

[3] D. Bertsekas, L. Polymenakos, and P. Tseng. An $\varepsilon$-relaxation method for separable convex cost network flow problems. *SIAM Journal on Optimization*, 7(3):853–870, 1997.

[4] E. Cohen and N. Megiddo. New algorithms for generalized network flows. *Mathematical Programming*, 64(1):325–336, 1994.

[5] G. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, 1963.

[6] N. Devanur, C. Papadimitriou, A. Saberi, and V. Vazirani. Market equilibrium via a primal–dual algorithm for a convex program. *Journal of the ACM (JACM)*, 55(5):22, 2008.

[7] J. Edmonds and R. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

[8] E. Eisenberg and D. Gale. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics*, 30(1):165–168, 1959.

[9] L. Fleischer and K. Wayne. Fast and simple approximation schemes for generalized flow. *Mathematical Programming*, 91(2):215–238, 2002.

[10] M. Fredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

[11] G. Goel and V. Vazirani. A perfect price discrimination market model with production, and a (rational) convex program for it. *Algorithmic Game Theory*, pages 186–197, 2010.

[12] A. Goldberg, S. Plotkin, and É. Tardos. Combinatorial algorithms for the generalized circulation problem. *Mathematics of Operations Research*, 16(2):351, 1991.

[13] A. Goldberg and R. Tarjan. Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM (JACM)*, 36(4):873–886, 1989.

[14] D. Goldfarb and Z. Jin. A faster combinatorial algorithm for the generalized circulation problem. *Mathematics of Operations Research*, pages 529–539, 1996.

[15] D. Goldfarb, Z. Jin, and J. Orlin. Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem. *Mathematics of Operations Research*, pages 793–802, 1997.

[16] D. Goldfarb and Y. Lin. Combinatorial interior point methods for generalized network flow problems. *Mathematical programming*, 93(2):227–246, 2002.

[17] D. Hochbaum and J. Shanthikumar. Convex separable optimization is not much harder than linear optimization. *Journal of the ACM (JACM)*, 37(4):843–862, 1990.

[18] K. Jain. Generalized online matching with concave utilities. *Unpublished manuscript*, 2011.

[19] K. Jain and V. Vazirani. Eisenberg-gale markets: Algorithms and game-theoretic properties. *Games and Economic Behavior*, 70(1):84–106, 2010.

[20] W. S. Jewell. Optimal flow through networks. *Operations Research*, 10:476–499, 1962.

[21] S. Kapoor and P. Vaidya. Speeding up Karmarkar's algorithm for multicommodity flows. *Mathematical programming*, 73(1):111–127, 1996.

[22] A. Karzanov and S. McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM J. Comput.*, 26(4):1245–1275, 1997.

[23] M. Minoux. Solving integer minimum cost flows with separable convex cost objective polynomially. *Netflow at Pisa*, pages 237–239, 1986.

[24] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press New York, NY, USA, 2007.

[25] K. Onaga. Dynamic programming of optimum flows in lossy communication nets. *IEEE Transactions on Circuit Theory*, pages 308–327, 1966.

[26] K. Onaga. Optimum flows in general communication networks. *Journal of the Franklin Institute*, 283(4):308–327, 1967.

[27] J. Orlin. Improved algorithms for computing fisher's market clearing prices: computing fisher's market clearing prices. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 291–300. ACM, 2010.

[28] J. B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations Research*, 41(2):338–350, 1993.

[29] T. Radzik. Approximate generalized circulation. *Technical Report93-2, Cornell Computational Optimization Project, Cornell University*, 1993.

[30] T. Radzik. Improving time bounds on maximum generalised flow computations by contracting the network. *Theoretical Computer Science*, 312(1):75–97, 2004.

[31] M. Restrepo and D. Williamson. A simple GAP-canceling algorithm for the generalized maximum flow problem. *Mathematical Programming*, 118(1):47–74, 2009.

[32] M. Shigeno. A survey of combinatorial maximum flow algorithms on a network with gains. *Journal of the Operations Research Society of Japan*, 47:244–264, 2004.

[33] M. Shigeno. Maximum network flows with concave gains. *Math. Program*, 107(3):439–459, 2006.

[34] V. I. Shmyrev. An algorithm for finding equilibrium in the linear exchange model with fixed budgets. *Journal of Applied and Industrial Mathematics*, 3(4):505–518, 2009.

[35] É. Tardos and K. Wayne. Simple maximum flow algorithms in lossy networks. In *Proceedings of 6th International Integer Programming and Combinatorial Optimization Conference, Lecture Notes in Computer Science*, volume 1412, pages 310–324, 1998.

[36] K. Truemper. On max flows with gains and pure min-cost flows. *SIAM Journal on Applied Mathematics*, 32(2):450–456, 1977.

[37] K. Truemper. Optimal flows in nonlinear gain networks. *Networks*, 8(1):17–36, 1978.

[38] V. V. Vazirani. The notion of a rational convex program, and an algorithm for the Arrow-Debreu Nash bargaining game. *Unpublished manuscript*, 2011.

[39] K. Wayne. A polynomial combinatorial algorithm for generalized minimum cost flow. *Mathematics of Operations Research*, pages 445–459, 2002.