INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 39, No. 2, May 2014, pp. 573–596
ISSN 0364-765X (print) | ISSN 1526-5471 (online)

http://dx.doi.org/10.1287/moor.2013.0623
© 2014 INFORMS

Concave Generalized Flows with Applications to Market Equilibria

László A. Végh
Department of Management, London School of Economics, London WC2A 2AE, United Kingdom, l.vegh@lse.ac.uk

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 [Truemper K (1978) Optimal flows in nonlinear gain networks. *Networks* 8(1):17–36] and by Shigeno [Shigeno M (2006) Maximum network flows with concave gains. *Math. Programming* 107(3):439–459]. We give a polynomial time combinatorial algorithm for solving corresponding flow maximization problems, finding an ε-approximate solution in $O(m(m\sigma+\log n)\log(MUm/\varepsilon))$ arithmetic operations, where M and U are upper bounds on simple parameters, and σ is the complexity of a value oracle query for the gain functions. This also gives a new algorithm for linear generalized flows, an efficient, purely scaling variant of the Fat-Path algorithm by Goldberg et al. [Goldberg AV, Plotkin SA, Tardos É (1991) Combinatorial algorithms for the generalized circulation problem. *Math. Oper. Res.* 16(2):351–381], not using any cycle cancellations.

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 provides combinatorial algorithms for various extensions of these market models. This includes nonsymmetric Arrow-Debreu Nash bargaining, settling an open question by Vazirani [Vazirani VV (2012) The notion of a rational convex program, and an algorithm for the Arrow-Debreu Nash bargaining game. *J. ACM* 59(2), Article 7].

Keywords: network flow algorithms; generalized flows; convex programming; market equilibrium
MSC2000 subject classification: Primary: 90C27; secondary: 90B10, 90C35, 90C25, 68R10, 91B50
OR/MS subject classification: Primary: networks/graphs: generalized networks; secondary: networks/graphs: flow algorithms; programming: nonlinear: algorithms; economics
History: Received July 24, 2012; revised February 28, 2013, and August 2, 2013. Published online in *Articles in Advance* September 30, 2013.

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 α units of flow enter arc e, then $\gamma_e\alpha$ units leave it. Since first studied by Kantorovich [23], Dantzig [5], and Jewell [21], the problem has found many applications in management science, including financial analysis and transportation; see Ahuja et al. [2, Chapter 15].

In this paper, we consider a nonlinear extension, *concave generalized flows*, studied by Truemper [40] in 1978, and by Shigeno [36] in 2006. For each arc e we are given a concave, monotone increasing function Γ_e such that if α units enter e then $\Gamma_e(\alpha)$ units leave it. We give a combinatorial algorithm for corresponding flow maximization 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 equilibrium settings, for which combinatorial algorithms have been developed over the last decade. As an application, we also get a combinatorial algorithm for nonsymmetric Arrow-Debreu Nash bargaining (ADNB), resolving an open question by Vazirani [42]. 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 such algorithm being the interior-point method by Kapoor and Vaidya [24]. Combinatorial approaches have been used since the 1960s (e.g., Jewell [21], Onaga [28], Truemper [39]), yet the first polynomial-time combinatorial algorithms were given only in 1991 by Goldberg et al. [14]. This inspired a line of research to develop further polynomial-time combinatorial algorithms, e.g., Cohen and Megiddo [4], Goldfarb and Jin [15], Goldfarb et al. [18], Tardos and Wayne [38], Fleischer and Wayne [10], Goldfarb et al. [17], Goldfarb and Lin [16], Wayne [44], Radzik [33], and Restrepo and Williamson [34]; for a survey on combinatorial generalized flow algorithms, see Shigeno [35]. Despite the vast literature, no strongly polynomial algorithm is known so far. Our algorithm for this special case derives from the Fat-Path algorithm in Goldberg et al. [14], with the remarkable difference that no cycle cancellations are needed.

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

Concave generalized flows being nonlinear convex programs, can also be solved by the ellipsoid method, yet no practically efficient methods are known for this problem. Hence finding a combinatorial algorithm is also a matter of running time efficiency. Shigeno [36] gave the first combinatorial algorithm that runs in polynomial time for some restricted classes of functions Γ_e, including piecewise linear. It is also an extension of the Fat-Path algorithm in Goldberg et al. [14]. In spite of this development, it has remained an open problem to find a combinatorial polynomial-time algorithm for arbitrary concave increasing gain functions.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

Our result settles this question by allowing arbitrary increasing concave gain functions provided via value oracle access. The running time bounds for this general problem are reasonably close to the most efficient linear generalized flow algorithms. Concave gain functions extend the applicability range of the classical generalized flow model, as they can describe, e.g., diminishing marginal utilities. From the application point of view, another contribution of the paper is extending generalized flow techniques to the domain of market equilibrium computations, where this model turns out to be a concise unifying framework.

The concave optimization problem might have irrational optimal solutions: in general, we give a fully polynomial-time approximation scheme, with running time dependent on $\log(1/\varepsilon)$ for finding an ε-approximate solution. In the market equilibrium applications we have rational convex programs (as in Vazirani [42]): 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 nonsymmetric Arrow-Debreu Nash bargaining, where the existence of a combinatorial algorithm was open (Vazirani [42]).

In §2, we give the precise definition of the problems considered. Thereby we introduce a new, equivalent variant of the problem, called the *symmetric formulation*, providing a more flexible algorithmic framework. 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. We first present our symmetric generalized flow algorithm in §5 for the special case of linear gains. Based on this, §6 gives the algorithm for arbitrary concave gain functions. Section 7 adapts these algorithms for the more standard sink formulation of the problems. 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. The final §9 discusses possible further directions.

2. Problem definitions. We define two closely related variants of the linear and the concave generalized flow problems. 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 do not allow parallel arcs and hence we may use ij to denote the arc from i to j. We also forbid oppositely directed pairs of arcs in the input. These restrictions are only for notational convenience and all results straightforwardly extend to a setting with parallel and oppositely directed arcs. We will use $V-t$ to denote the set $V\setminus\{t\}$.

In the linear setting, we are given lower and upper arc capacities $l, u\colon E\to\mathbb{R}$ and gain factors $\gamma\colon E\to\mathbb{R}^+$ on the arcs, and node demands $b\colon V\to\mathbb{R}$. By a *pseudoflow* we mean a function $f\colon E\to\mathbb{R}$ with $l\le f\le u$. 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 *sink* node $t\in V$. The objective is to maximize e_t for pseudoflows satisfying $e_i\ge 0$ 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 $l\equiv 0$, $b\equiv 0$. However, this problem can easily be reduced to solving the sink formulation; see e.g., Shigeno [35].

The convex extension has been proposed by Truemper [40] and Shigeno [36]. On each arc $ij\in E$, we are given lower and upper arc capacities $l, u\colon E\to\mathbb{R}$ and a monotone increasing continuous concave function $\Gamma_{ij}\colon [l_{ij}, u_{ij}]\to\mathbb{R}\cup\{-\infty\}$; we are also given node demands $b\colon V\to\mathbb{R}$. As for generalized flows, a pseudoflow is a function $f\colon E\to\mathbb{R}$ with $l\le f\le u$. 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*, we say that the pseudoflow f is *feasible*, if $e_i\ge 0$ for all $i\in V-t$ and $e_t>-\infty$. The objective is to maximize e_t for feasible pseudoflows.

Shigeno [36] defines this problem with $e_i=0$ if $i\in V-t$, and $b\equiv 0$ and without explicit capacity constraints. She also discusses the version with $e_i\ge 0$ and gives a reduction from the original version to this one. Whereas capacity constraints can be simulated by the functions Γ_{ij}, we impose them explicitly as they will be included in the running time bounds. The formulation with $e_i\ge 0$ seems more natural as it gives a convex optimization problem, which is not the case for $e_i=0$. Observe that we allow $\Gamma_{ij}(f_{ij})=-\infty$; the reason is that we use the gain function $\Gamma_{ij}(\alpha)=c_{ij}\log\alpha$ on certain arcs in the market equilibrium applications. Having $\Gamma_{ij}(f_{ij})=-\infty$ on any arc implies $e_j=-\infty$ and thus contradicts feasibility; therefore we must enforce higher f_{ij} values on such arcs.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

In the sink formulation, the node t plays a distinguished role. It turns out to be more convenient 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 $e_i \geq 0$ 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 $\kappa_i \geq 0$. The objective is to minimize $\kappa_f = \sum_{i \in V} M_i \kappa_i$ for a pseudoflow f subject to $e_i + \kappa_i \geq 0$ for each $i \in V$.

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

The sink version fits into this framework with $M_i = \infty$ for $i \neq t$, $M_t = 1$ and $b_t = \infty$. As shown in §7, we can set finite, polynomially bounded M_i, and b_t values, such that the symmetric version returns an optimal (or sufficiently close approximate) solution to the sink version, both for linear and for concave gain functions. Besides the sink version, another natural setting is when $M_i = 1$ for all $i \in V$, that is, maintaining $e_i \geq 0$ has the same importance for all nodes.

Although the symmetric formulation could seem more general than 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 sink 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 its pertinence to our algorithmic purposes.

2.1. Complexity model. The complexity setting will be different for generalized flows and for concave generalized flows. For generalized flows, we aim to find an optimal solution, and in the concave case, only an approximate one. For generalized flows, the gain functions are given explicitly as linear functions, and in the concave case, the description of the functions might be infinite. To handle this difficulty, following the approach of Hochbaum and Shanthikumar [19], we assume oracle access to the Γ_{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 $\alpha \in [l_{ij}, u_{ij}]$; and (ii) inverse value oracle, returning a value β with $\alpha = \Gamma_{ij}(\beta)$ for any $\alpha \in [\Gamma_{ij}(l_{ij}), \Gamma_{ij}(u_{ij})]$.

We assume that both oracles return the exact (possibly irrational) solution, and any oracle query is done in time σ. 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 results naturally extend to the setting with only approximate oracles and computational capacities in a straightforward manner. Notice that in an approximate sense, an inverse value oracle can be simulated by a value oracle.

By an *ε-approximate solution* to the symmetric concave generalized flow problem we mean a feasible solution with the excess discrepancy larger than the optimum by at most ε. An *ε-approximate solution* to the sink version means a pseudoflow with the objective value e_t at most ε less than the optimum, and the total violation of the inequalities $e_i \geq 0$ for $i \in V - t$ is also at most ε. (Note that an ε-approximate solution is thus not necessarily feasible.)

In the symmetric formulation of both the linear and concave generalized flow problems, we assume that all M_i values are positive integers, and let M denote their maximum.

For generalized flows, we assume all l, u, and b values are given as integers and the γ values as rational numbers; let B be the largest integer used in their descriptions. The running time bound will be $O(m^2(m \log B + \log M) \log n)$ for the symmetric formulation and $O(m^2(m + n \log n) \log B)$ for the sink formulation. This is the same as the complexity bound of the highest gain augmenting path algorithm by Goldfarb et al. [18]. The best current running time bounds are $O(m^{1.5} n^2 \log B)$ using an interior point approach by Kapoor and Vaidya [24] and $\tilde{O}(m^2 n \log B)$[1] by Radzik [33], that is an enhanced version of Goldfarb et al. [18].

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 l_{ij}, u_{ij}, and the $\Gamma_{ij}(l_{ij})$, $\Gamma_{ij}(u_{ij})$ values. For each arc ij, let us define $r_{ij} = |\Gamma_{ij}(l_{ij})|$ whenever $\Gamma_{ij}(l_{ij}) > -\infty$ and $r_{ij} = 0$ otherwise. Let

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

For the sink version, we need to introduce one further complexity parameter U^* because of difficulties arising if $\Gamma_{ij}(l_{ij}) = -\infty$ for certain arcs. Let U^* satisfy $U \leq U^*$, and $e_t \leq U^*$ for any pseudoflow (it is easy to see that

[1] The $\tilde{O}()$ notation hides a polylogarithmic factor.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

$U^* = d_t U$ 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 $i \in V - t$ and $e_t > -\infty$), there exists one with $e_t \geq -U^*$. If $\Gamma_{jt}(l_{jt}) > -\infty$ for each arc $jt \in E$, then $U^* = d_t U$ also satisfies this property. For the case when $-\infty$ values are allowed, a bound on U^* can be given as in §8.

The main result is as follows:

THEOREM 1. *For the symmetric formulation of the concave generalized flow problem, there exists a combinatorial algorithm that finds an ε-approximate solution in running time $O(m(m\sigma + n\log n)\log(MUm/\varepsilon))$. For the sink formulation, there exists a combinatorial algorithm that finds an ε-approximate solution in $O(m(m + n\log n)\log(U^*m/\varepsilon))$ time. In both cases, the running time bound is on the number of arithmetic operations and oracle queries.*

The starting point of our investigation is the Fat-Path algorithm in Goldberg et al. [14]. The first important idea is using the symmetric formulation. This is a more flexible framework, and thus we will be able to entirely avoid cycle cancellation and use excess transportation phases only. Our (linear) generalized flow algorithm is the first generalized flow algorithm that uses a pure scaling technique, without any cycle cancellation. The key new idea here is the way "Δ-positive" and "Δ-negative" nodes are defined, maintaining a "security reserve" in each node that compensates for adjustments when moving from the Δ-scaling phase to the $\Delta/2$-phase.

We extend the linear algorithm to the concave setting using a local linear approximation of the gain functions, following Shigeno [36]. This approximation is motivated by the technique of Minoux [26] and Hochbaum and Shanthikumar [19] for minimum cost flows with separable convex objectives.

3. Applications to market equilibrium and Nash bargaining problems. Intensive research has been pursued over the last decade to develop polynomial-time combinatorial algorithms for certain market equilibrium problems. The starting point is the algorithm by Devanur et al. [7] for computing market clearing prices in Fisher's model with linear utilities, followed by a study of several variants and extensions of this model. For a survey, see Nisan et al. [27, Chapter 5] or Vazirani [42].

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 $j \in G$, if i buys x_{ij} units of good j, then she accrues $U_{ij}x_{ij}$ utility for some $U_{ij} \geq 0$. Let $n = |B| + |G|$ and m be the number of pairs ij with $U_{ij} > 0$. We assume there is such an edge incident to every buyer and to every good. 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 equilibrium solutions for linear Fisher markets were described via the convex program (EG) by Eisenberg and Gale [9] in 1959; the combinatorial algorithms for this problem and other models rely on the Karush-Kuhn-Tucker (KKT)-conditions for the corresponding convex programs. Exact optimal solutions can be found, since these problems admit rational optimal solutions:

$$\begin{gathered}\max \sum_{i\in B} m_i \log z_i \\ z_i \leq \sum_{j\in G} U_{ij}x_{ij} \quad \forall\, i \in B \\ \sum_{i\in B} x_{ij} \leq 1 \quad \forall\, j \in G \\ z, \quad x \geq 0.\end{gathered} \tag{EG}$$

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

For the Eisenberg-Gale program, let us define the graph (V, E) with $V = B \cup G \cup \{t\}$. Let $ji \in E$ whenever $j \in G$, $i \in B$, $U_{ij} > 0$, and set $\Gamma_{ji}(\alpha) = U_{ij}\alpha$ as a linear gain function. Also, let $it \in E$ for every $i \in B$ 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 set of feasible solutions.) Hence our general algorithm gives an ε-approximation for this problem. In §8, we show that for sufficiently small ε we can transform it to an exact optimal solution.

INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

The flexibility of the concave generalized flow model enables various extensions. For example, we can replace each linear function $U_{ij}\alpha$ by an arbitrary concave increasing function, obtaining the perfect price discrimination model of Goel and Vazirani [12]. They studied piecewise linear utility functions; our model enables arbitrary functions (although the optimal solution may be irrational).

In the *Arrow-Debreu Nash bargaining* (*ADNB*) defined by Vazirani [42], traders arrive to the market with initial endowments of goods, giving utility c_i for player i. They want to redistribute the goods to obtain higher utilities using Nash bargaining. The disagreement point is when everyone keeps the initial endowment, guaranteeing her $c_i \geq 0$ utility. In an optimal Nash bargaining solution we maximize $\sum_{i\in B}\log(z_i - c_i)$ over the constraint set in (EG). Unlike for the linear Fisher model, equilibrium prices may not exist, corresponding to a disagreement solution. A sophisticated two phase algorithm is given in Vazirani [42], first for deciding feasibility, then for finding the equilibrium solution.

The convex program for ADNB can be obtained from the Eisenberg-Gale program by modifying the objective to $\sum_{i\in B}\log(z_i - c_i)$ and adding $z_i \geq c_i$ for every $i \in B$. In the formulation as a concave generalized flow, this corresponds to modifying the gain function on the it arcs to $\Gamma_{it}(\alpha) = \log(\alpha - c_i)$ and setting the lower capacity to c_i. Hence this problem also fits into our framework. From this general perspective, it does not seem more difficult than the linear Fisher model.

Nonsymmetric Nash bargaining was defined by Kalai [22]. For ADNB, it corresponds to maximizing $\sum_{i\in B} m_i \log(z_i - c_i)$ over the constraint set in (EG), for some positive coefficients m_i. The algorithm in Vazirani [42] heavily relies on the assumption $m_i = 1$, and does not extend to this more general setting, called *nonsymmetric ADNB*. Finding a combinatorial algorithm for this latter problem was left open in Vazirani [42]. Another open question in Vazirani [42] 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.

We assume all values U_{ij}, m_i, and c_i are integers. Let $U_{\max} = \max\{U_{ij}\colon i \in B, j \in G\}$, $R = \max\{m_i\colon i \in B\}$, and $C = \max c_i$. In §8, we show how our algorithm can be used to find an exact solution to the nonsymmetric ADNB problem in time $O(m(m + n\log n)(n\log(nU_{\max}R) + \log C))$. The running time bound in Vazirani [42] for symmetric ADNB ($R = 1$) is $O(n^8 \log U_{\max} + n^4 \log C)$.

Let us also remark that an alternative convex program for the linear Fisher market, given by Shmyrev [37], shows that it also fits into the framework of minimum-cost circulations with a separable convex cost function, and thus can be solved by the algorithms of Hochbaum and Shanthikumar [19] or Karzanov and McCormick [25]. Recently, Végh [43] gave a strongly polynomial algorithm for a class of these problems, which includes Fisher's market with linear utilities and also a generalization called spending constraint utilities. However, this does not seem to capture perfect price discrimination or ADNB, where no alternative formulations analogous to Shmyrev [37] are known.

As further applications of the concave generalized flow model, we can take single-source multiple-sink markets by Jain and Vazirani [20], or concave cost matchings studied by Devanur and Jain [6].

A distinct characteristic of the Eisenberg-Gale program and its extensions is that they are rational convex programs. We may lose this property when changing to general concave spending constraint utilities. However, 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 7 shows that under certain technical conditions, our approximation algorithm can be turned into a polynomial time algorithm for finding an exact optimal solution. We shall verify these conditions for nonsymmetric ADNB.

4. Background and overview. The minimum-cost circulation problem[2] is fundamental to all problems and algorithms discussed in the paper. We give an overview in §4.1. We present the two main algorithmic paradigms: cycle cancelling and successive shortest paths along with their efficient variants. As already revealed by early studies of the problem (e.g., Onaga [29] and Truemper [39]), 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. In §4.2, we continue with an overview of generalized flow algorithms, exhibiting some important ideas and their relation to minimum-cost circulations. We also exhibit here the main ideas of our algorithm for the linear case. 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 §4.4 we consider the concave generalized flow problem and discuss the algorithm by Shigeno [36] and its relation to algorithmic ideas of the previous problems. We emphasize some difficulties and outline the ideas of our solution.

[2] We shall use the term "circulation" to distinguish from other flow problems in the paper.

INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

4.1. Minimum-cost flows: Cycle cancelling and successive shortest paths. In the minimum-cost circulation problem, given is a directed graph $G=(V,E)$ with lower and upper arc capacities $l,u\colon E\to\mathbb{R}\cup\{\infty\}$, costs $c\colon E\to\mathbb{R}$ on the arcs, and node demands $b\colon V\to\mathbb{R}$ with $\sum_{i\in V} b_i=0$. Let

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

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

Linear programming duality 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 $l_{ji}<f_{ji}$. The first types of arcs are called *forward arcs* and are assigned the original cost c_{ij}, while the latter arcs are *backward arcs* assigned 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 potential $\pi\colon V\to\mathbb{R}$ 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., Ahuja et al. [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; such a cycle can be found efficiently. Sending some flow around this cycle decreases the objective and maintains feasibility, providing the next solution.

In the *successive shortest path framework* (see, e.g., Ahuja et al. [2, Chapter 9.7]), we waive feasibility by allowing $e_i>0$ or $e_i<0$; 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 $ij\in E_f$, with ε being equal to the negative of the minimum mean cycle value, decreasing exponentially over time.

Polynomial implementations of the successive shortest path algorithm can be obtained by capacity scaling; the most efficient, strongly polynomial such algorithm is due to Orlin [30]. We describe here a basic capacity scaling framework by Edmonds and Karp [8], which was the first (weakly) polynomial algorithm for the problem. Instead of the residual graph E_f, we consider the Δ-residual graph $E_f(\Delta)$ consisting of arcs with residual capacity at least Δ (the residual capacity is $u_{ij}-f_{ij}$ on a forward arc ij and $f_{ji}-l_{ji}$ on a backward arc). The algorithm consists of Δ-scaling phases, with Δ decreasing by a factor of 2 between two phases. In a Δ-phase, we iteratively send Δ 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 in $E_f(\Delta)$. The Δ-phase finishes when this is no longer possible, which means the total positive excess is at most $n\Delta$.

In the Δ-phase, $\pi_j-\pi_i\le c_{ij}$ is maintained on arcs of the Δ-residual graph. When moving to the $\Delta/2$ phase, this might not hold anymore, since the $\Delta/2$-residual graph contains more arcs, namely, the ones with residual capacity between $\Delta/2$ and Δ. 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. Linear generalized flows—cycle cancelling and excess transportation. In what follows, we consider the sink version of the generalized flow problem, with sink $t\in V$. For a pseudoflow $f\colon E\to\mathbb{R}$, 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 in E_f. We can modify f by sending some flow $\alpha>0$ around C from some $i\in V(C)$. This leaves e_j unchanged if $j\ne i$, and increases e_i by $(\gamma(C)-1)\alpha$, where $\gamma(C)=\Pi_{e\in C}\gamma_e$. If $\gamma(C)>1$ then we call C a *flow-generating cycle*, and for $\gamma(C)<1$, a *flow-absorbing cycle*, since we can generate or eliminate excess at an arbitrary node $i\in C$, respectively. The amount of flow that can be generated is of course bounded by the capacity constraints.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

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

The gain factors γ_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 μ: $V \to \mathbb{R}_{>0} \cup \{\infty\}$ with $\mu_t = 1$ a label function. *Relabeling* the pseudoflow f by μ means dividing the flow on each arc ij going out from i by μ_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 $\gamma^\mu_{ij} \le 1$ for all $ij \in E_f$, that is, no arc may increase the relabeled flow.

Assume we have a conservative labeling μ 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 may exist, and the converse can also be shown to hold. Note that on V', $\pi_i = -\log \mu_i$ is a feasible potential for $c_e = -\log \gamma_e$ if and only if μ 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 indeed be implemented in strongly polynomial time; see Radzik [32] and Shigeno [35]. The additional 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 [29], 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 successive 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 et al. [14]. One of them, Fat-Path, uses a method analogous to capacity scaling. A path P in E_f from a node i to t is called Δ-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 Δ.

The algorithm consists of Δ-phases, with Δ decreasing by a factor of 2 for the next phase. In the Δ-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 Δ-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 Δ-phase, $e^*_t - e_t \le 2(n+m)\Delta$ for the optimum value e^*_t and thus the number of path augmentations in each Δ-phase can be bounded by $2(n+m)$. Arriving at a sufficiently small value of Δ, it is possible to obtain an optimal solution by a single maximum flow computation.

The basic framework of Onaga [29] and of Fat-Path, namely, using different subroutines for eliminating flow-generating cycles and for transporting excess to the sink, has been adopted by most subsequent algorithms, e.g., Goldfarb and Jin [15], Goldfarb et al. [18], Tardos and Wayne [38], Fleischer and Wayne [10], and Radzik [33]. Among them, Goldfarb et al. [18] is an almost purely scaling polynomial time algorithm, but it still needs an initial cycle-cancelling phase as in Onaga [29].

In contrast, our algorithm does not need any cycle cancelling, and adapts Fat-Path to a pure successive shortest paths framework. The successive shortest paths algorithms for minimum-cost circulations start with an infeasible pseudoflow, having both positive and negative nodes. To use an analogous method for generalized flows, we have to give up the standard framework of algorithms where $e_i \ge 0$ 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.

Similar to Fat-Path, we use a scaling algorithm. In the Δ-phase, we consider the residual graph restricted to Δ-fat arcs, arcs that may participate in a highest gain Δ-fat-path, and maintain a conservative labeling μ with $\gamma^\mu_{ij} \le 1$ on the Δ-fat arcs. When moving to the $\Delta/2$-phase, this condition may get violated because of $\Delta/2$-fat arcs that were not Δ-fat. Analogously to the Edmonds-Karp algorithm, we modify the flow by saturating each violated arc and thereby restitute dual feasibility. However, these changes may create new negative nodes and thus violate the condition $\mu_i = 1/M_i$ for $e_i < 0$, which we must maintain.

We resolve this difficulty by maintaining a "security reserve" of $d_i\Delta\mu_i$ in each node i (d_i is the number of incident arcs). This gives an upper bound on the total change caused by restoring feasibility of incident arcs

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

in all subsequent phases. We call a node Δ-positive if $e_i > d_i\Delta\mu_i$, Δ-negative if $e_i < d_i\Delta\mu_i$, and Δ-neutral if $e_i = d_i\Delta\mu_i$. Δ-negative nodes may become negative ($e_i < 0$) at a later phase, and therefore we maintain the stronger condition $\mu_i = 1/M_i$ for them. We send flow from Δ-positive nodes to Δ-negative and Δ-neutral ones. Thereby we treat some nodes with $e_i > 0$ as sinks and increase their excess further; however, as Δ decreases, such nodes may gradually become sources.

For the sink version, described in §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. We remark that the highest gain path algorithm (Goldfarb et al. [18]) can also be modified to a purely scaling algorithm using the symmetric formulation that enables to start from an arbitrary nonfeasible solution and thereby eliminate the initial cycle-cancelling phase.

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_{ij} by a convex function $C_{ij}(\cdot)$. We are given a directed graph $G = (V, E)$ with lower and upper arc capacities l, u: $E \to \mathbb{R}$, convex cost functions C_{ij}: $[l_{ij}, u_{ij}] \to \mathbb{R}$ on the arcs, and node demands b: $V \to \mathbb{R}$ with $\sum_{i\in V} b_i = 0$. Our aim is to minimize $\sum_{ij\in E} C_{ij}(f_{ij})$ for feasible circulations f. This is a widely applicable framework; see Ahuja et al. [2, Chapter 14].

This is a convex optimization problem, and optimality can be described by the KKT conditions. Let $C^+_{ij}(\alpha)$ denote the left derivative of C_{ij}. 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; f is optimal if and only if there exists no cycle C in E_f with $\sum_{ij\in C} C^+_{ij}(f_{ij}) < 0$. In dual terms, f is optimal if and only if there exists a potential π: $V \to \mathbb{R}$ with $\pi_j - \pi_i \le C^+_{ij}(f_{ij})$ for all $ij \in E_f$.

Both the minimum mean cycle cancellation 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 cancellation was adapted by Karzanov and McCormick [25]. The algorithm subsequently cancels cycles in E_f with minimum mean value with respect to the $C^+_{ij}(f_{ij})$ 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

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

for the current potential π and scaling parameter ε.

For capacity scaling, Minoux [26] used the following approach, that was later extended further by Hochbaum and Shanthikumar [19]; see also Ahuja et al. [2, Chapter 14.5]. The algorithm consists of Δ-phases. In the Δ-phase, each C_{ij} is linearized with granularity Δ.

Let $E_f(\Delta)$ denote the Δ-residual network. We will maintain Δ-optimality, that is, there exists a potential π such that

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

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

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

Recently, Végh [43] gave a strongly polynomial capacity scaling algorithm for a class of objective functions, including convex quadratic objectives, and Fisher's market with linear and with spending constraint utilities (based on Shmyrev's formulation Shmyrev [37]). Let us also remark that the results of Karzanov and McCormick [25] and Hochbaum and Shanthikumar [19] actually address much more general problems: minimizing convex objectives over polyhedra given by matrices with bounded subdeterminants. The framework of Hochbaum and Shanthikumar [19] needs weaker assumptions on the objective function and on the oracle.

4.4. Concave generalized flows. As we have seen, both the cycle cancelling and capacity scaling approaches for minimum-cost circulations naturally extend to separable convex cost functions. Similarly, our algorithm in §6 for concave gain functions is a natural extension of the generalized flow algorithm in §5.

Nevertheless, we were not able to extend any of the previous generalized flow algorithms to concave gains. Shigeno's [36] approach was to extend the Fat-Path algorithm of Goldberg et al. [14]. However, Shigeno [36]

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

obtains polynomial running time bounds only for restricted classes of gain functions. The algorithm consists of two procedures applied alternately similar to Fat-Path: a cycle cancellation phase to generate excess on cycles with positive gains, and a path augmentation phase to transport new excess to the sink in chunks of Δ. For both phases, previous methods naturally extend: cycle cancelling is performed analogously to Karzanov and McCormick [25], whereas path augmentation to Hochbaum and Shanthikumar [19]. 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. Whereas for generalized flows, a cycle-cancelling phase completely eliminates flow-generating cycles, here we can only get an approximate solution allowing some small positive gain on cycles at termination. 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 Δ. Yet if some small error $\varepsilon > 0$ is allowed, then no general guarantee can be given so that (3) holds for a certain value Δ.

For this reason, our goal was to avoid using the two different frameworks simultaneously. It turns out that our scaling-type linear generalized flow algorithm (outlined in §4.2) smoothly extends to this general setting. We use the local linearization $\theta^{\mu}_{\Delta}(ij)$ of Γ_{ij} used by Shigeno [36], an analogue of (3). In the Δ-phase, we consider the graph of Δ-fat arcs, and maintain $\theta^{\mu}_{\Delta}(ij) \le 1$ on them.

When moving from a Δ-phase to a $\Delta/2$-phase in the linear algorithm, the only reason for infeasibility is due to $\Delta/2$-fat arcs that were not Δ-fat. In contrast, feasibility can be violated on Δ-fat arcs as well, as $\theta^{\mu}_{\Delta}(ij) \le 1 < \theta^{\mu}_{\Delta/2}(ij)$ may happen because of the finer linear approximation of the gain functions in the $\Delta/2$-phase. Fortunately, feasibility can be restored in this case as well, by changing the flow on each arc by a small amount.

5. Linear generalized flow 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 incorporate the sink version. However, in the algorithmic parts, we restrict ourselves to finite M_i values.

We describe optimality conditions in §5.1. Concepts and results here are well known in the generalized flow literature, thus we do not include references. Section 5.2 introduces Δ-fat arcs and Δ-conservative labelings, the feasibility framework in the Δ-phase. Section 5.3 describes the subroutine Tighten-Label, an adaptation of Dijkstra's algorithm that finds highest gain Δ-fat paths. The main algorithm is exhibited in §5.4. Analysis and running time bounds are given in §5.5. The final step of the algorithm is deferred to §5.6, where we show that when the total relabeled excess is sufficiently small, an optimal solution can be found by a single maximum flow computation.

5.1. Optimality conditions.

Let $G = (V, E)$ be a network with lower and upper capacities l, u, gain factors γ, node demands b, and penalty factors M. In the sequel, we assume all lower capacities are 0. Every problem instance of symmetric generalized flows can be simply transformed to an equivalent one in this form in the following way. For each arc $ij \in E$, increase the node demand b_i by l_{ij} and decrease b_j by $\gamma_{ij} l_{ij}$. Modify the lower capacity of ij to 0 and the upper to $u_{ij} - l_{ij}$.

For 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 $f_{ji} > 0$. The first type of arcs are called *forward arcs*, and the second type are the *backward arcs*. (Recall that we have forbidden pairs of oppositely directed arcs in the input, and hence these notions are well defined.) For a forward arc ij, let γ_{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 $ji \in E_f$. For backward arcs, the capacities are $l_{ji} = -\gamma_{ji} u_{ji}$ and $u_{ji} = 0$. By increasing (decreasing) f_{ji} by α, we mean decreasing (increasing) f_{ij} by α/γ_{ij}.

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

Let $C = i_0, \ldots, i_{k-1} i_0$ be a cycle in E_f. Sending α units of flow on C from i_0 modifies only e_{i_0}, increasing by $(\gamma(C) - 1)\alpha$ for $\gamma(C) = \prod_{e \in C} \gamma_e$; 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 α units around (assuming that α is sufficiently small so that no capacity constraints are violated).

The pair (C, P) is called a *generalized augmenting path* 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 ($i_0 = t$, $P = \varnothing$ is possible);

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

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

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

Note that GAPs were already defined in §4.2 for the sink version of the problem. The definition here is slightly different because we do not have a distinguished sink anymore; in particular, type (c) GAPs do not appear in the sink version.

Lemma 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 values of α, 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. □

The dual description of an optimal solution is in terms of *relabelings* with a label function $\mu\colon V \to \mathbb{R}_{>0} \cup \{\infty\}$. For each node $i \in V$, let us rescale the flow on all arcs $ij \in E$ by μ_i: let $f_{ij}^\mu := f_{ij}/\mu_i$. We get a problem equivalent to the original one with relabeled gains $\gamma_{ij}^\mu := \gamma_{ij}\mu_i/\mu_j$. Accordingly, the relabeled demands, excesses, and capacities are $b_i^\mu := b_i/\mu_i$, $e_i^\mu := e_i/\mu_i$, and $u_{ij}^\mu := u_{ij}/\mu_i$. A relabeling is *conservative*, if for any residual arc $ij \in E_f$, $\gamma_{ij}^\mu \le 1$, that is, no arc may increase the relabeled flow. Furthermore, for each $i \in V$, $\mu_i \ge 1/M_i$ is required and 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 $b_i^\mu = e_i^\mu = 0$, and $\gamma_{ji}^\mu = 0$ for all arcs $ji \in E_f$. If $ij \in E_f$, $\mu_i = \infty$, and $\mu_j < \infty$, then $\gamma_{ij}^\mu = \infty$. Consequently, if μ 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 μ is conservative, there exists no flow generating cycle on the node set $\{i\colon \mu_i < \infty\}$. This is because for a cycle C, $\gamma(C) = \prod_{ij\in C}\gamma_{ij} = \prod_{ij\in C}\gamma_{ij}^\mu$.

Theorem 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 μ with $e_i = 0$ whenever $1/M_i < \mu_i < \infty$.*

Proof. The equivalence of (i) and (iii) is by linear programming duality, with μ_i being the reciprocal of the dual variable corresponding to the inequality $e_i + \kappa_i \ge 0$; (i) implies (ii) by Lemma 1.

It is left to show that (ii) implies (iii). If the excess discrepancy is 0 (that is, $e_i \ge 0$ for all $i \in V$), then $\mu \equiv \infty$ is conservative. Otherwise, let $N = \{t\colon e_t < 0\}$. 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 μ_i be the smallest possible value of $1/(\gamma(P)M_t)$ for $\gamma(P) = \prod_{e\in P}\gamma_e$, 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 $\gamma(P)$.

The relabeling μ clearly satisfies $\gamma_{ij}\mu_i/\mu_j \le 1$. We shall prove $\mu_i \ge 1/M_i$ for each $i \in V$ and $\mu_t = 1/M_t$ for each $t \in N$. If $e_i > 0$, then no such path P may exists as it would give a type (b) GAP. Consequently, $\mu_i = \infty$. If $e_i \le 0$, then $\mu_i \ge 1/M_i$ as otherwise the optimal P defining μ_i would be a type (c) GAP. Finally, if $t \in N$, then $\mu_t \le 1/M_t$ as the gain of the path $P = \varnothing$ is defined as 1. □

5.2. Δ-conservative labels. The residual capacity of arc $ij \in E_f$ is $u_{ij} - f_{ij}$ (for a backward arc ij, this is $\gamma_{ji}f_{ji}$). In contrast, we define the *fatness* of $ij \in E_f$ by $s_f(ij) := \gamma_{ij}(u_{ij} - f_{ij})$ (on backward arcs, $s_f(ij) = f_{ji}$). The fatness expresses the maximum possible flow increase in j if we saturate ij. This notion enables us to identify arcs that can participite in fat paths during the algorithm. In accordance with the other variables, the relabeled fatness is defined as $s_f^\mu(ij) := s_f(ij)/\mu_j$.

For the scaling parameter $\Delta > 0$, we define the following relaxation of conservativeness. Let $\mu\colon V \to \mathbb{R}_{>0} \cup \{\infty\}$ be a label function. Recall that d_i is the total number of arcs incident to i. A node $i \in V$ is called Δ-*negative* if $e_i^\mu < d_i\Delta$, Δ-*neutral* if $e_i^\mu = d_i\Delta$, and Δ-*positive* if $e_i^\mu > d_i\Delta$.

The Δ-*fat graph* $E_f^\mu(\Delta)$ is the set of residual arcs of relabeled fatness at least Δ:

$$E_f^\mu(\Delta) := \{ij \in E_f\colon s_f^\mu(ij) \ge \Delta\}.$$

Arcs in $E_f^\mu(\Delta)$ will be called Δ-*fat arcs*. The labeling μ is called Δ-*conservative*, if $\gamma_{ij}^\mu \le 1$ holds for every $ij \in E_f^\mu(\Delta)$, and $\mu_i \ge 1/M_i$ for all $i \in V$. Further, for every Δ-negative node i, we require $\mu_i = 1/M_i$.

By a 0-conservative labeling we mean a conservative one. An important difference between conservative and Δ-conservative labelings for $\Delta > 0$ is that nodes with $\mu_i = \infty$ may not be present in the latter one, because of the constraint on Δ-negative nodes. Let $Ex^\mu(f) := \sum_{i\in V}\max\{e_i^\mu, 0\}$ and $Ex_\Delta^\mu(f) := \sum_{i\in V}\max\{e_i^\mu - d_i\Delta, 0\}$ denote the total relabeled excess and total modified relabeled excess for Δ, respectively. Note that $Ex^\mu(f) \le Ex_\Delta^\mu(f) + 2m\Delta$.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

LEMMA 2. *Let f be a pseudoflow with a Δ-conservative labeling μ. Let $0 \le \Delta' < \Delta$. Then there exists a flow $\bar{f}$ such that μ is Δ'-conservative for $\bar{f}$ and $Ex^{\mu}_{\Delta'}(\bar{f}) \le Ex^{\mu}_{\Delta}(f) + 3m(\Delta - \Delta')$.*

PROOF. We shall construct $\bar{f}$ by modifying f on each arc independently. For Δ-fat arcs, Δ-conservativeness guarantees $\gamma^{\mu}_{ij} \le 1$. Consider an arc $ij \in E^{\mu}_f(\Delta') - E^{\mu}_f(\Delta)$, that is,

$$\Delta' \le \gamma^{\mu}_{ij}(u^{\mu}_{ij} - f^{\mu}_{ij}) < \Delta,$$

and assume $\gamma^{\mu}_{ij} > 1$. Let us set $\bar{f}_{ij} = \min\{u_{ij}, f_{ij} + ((\Delta - \Delta')\mu_j)/\gamma_{ij}\}$. Then ij cannot be Δ'-fat, since either $ij \notin E_f$, or $\gamma^{\mu}_{ij}(u^{\mu}_{ij} - \bar{f}^{\mu}_{ij}) < \Delta'$.

We have $f^{\mu}_{ji} - \bar{f}^{\mu}_{ji} = \gamma^{\mu}_{ij}(\bar{f}^{\mu}_{ij} - f^{\mu}_{ij}) \le \Delta - \Delta'$. Also, $\bar{f}^{\mu}_{ij} - f^{\mu}_{ij} \le \Delta - \Delta'$ holds because of $\gamma^{\mu}_{ij} > 1$. We also have to consider the possibility that ji is Δ'-fat for $\bar{f}$. In this case, conservativeness is guaranteed since $\gamma^{\mu}_{ji} = 1/\gamma^{\mu}_{ij} < 1$.

To complete the proof of Δ'-conservativeness, we show that $\bar{f}$ has no Δ'-negative nodes with $\mu_i > 1/M_i$. As we have seen, both f^{μ}_{ij} and $\gamma^{\mu}_{ij} f^{\mu}_{ij}$ change by at most $\Delta - \Delta'$. Consequently for every $i \in V$, the total possible change of the relabeled flow on arcs incident to i is $d_i(\Delta - \Delta')$. A node is nonnegative for Δ if $e^{\mu}_i \ge d_i\Delta$ and for Δ' if $e^{\mu}_i \ge d_i\Delta'$. Therefore, a Δ-nonnegative node cannot become Δ'-negative, proving the claim.

Further, $Ex^{\mu}_{\Delta'}(f) \le Ex^{\mu}_{\Delta}(f) + \sum_{i\in V} d_i(\Delta - \Delta')$, and on each arc, at most $\Delta - \Delta'$ units of new excess is created. This gives $Ex^{\mu}_{\Delta'}(\bar{f}) \le Ex^{\mu}_{\Delta}(f) + 3m(\Delta - \Delta')$. $\square$

The proof also gives a straightforward algorithm for finding such an $\bar{f}$. Let Adjust(Δ, Δ') denote this subroutine. In particular, Adjust$(\Delta, 0)$ finds an $\bar{f}$ for which μ is a conservative labeling. Further, if there are no Δ-negative nodes for f and μ, then $\bar{f}$ is a 0-discrepancy optimal solution.

5.3. Δ-canonical labels. An arc ij is called *tight* if $\gamma^{\mu}_{ij} = 1$, and a directed path is *tight* if it consists of tight arcs. Given a pseudoflow f, a conservative labeling μ is called *canonical*, if for each $i \in V$ with $\mu_i < \infty$, there exists a tight path in E_f from i to a negative node. Analogously, for $\Delta > 0$, a labeling is called Δ-*canonical*, if it is Δ-conservative, and for each $i \in V$ there exists a tight path in $E^{\mu}_f(\Delta)$ from i to some Δ-negative or Δ-neutral node. Such a path is a highest gain Δ-fat path as in Goldberg et al. [14]. (Note that $\mu_i < \infty$ for every $i \in V$ for Δ-canonical labelings, and also that paths are allowed to end in Δ-neutral nodes, in contrast to canonical labelings.) By 0-canonical labeling we mean a canonical one.

Given a Δ-conservative relabeling μ that is not canonical, Tighten-Label(f, μ, Δ) replaces μ by a Δ-canonical labeling μ' with $\mu'_i \ge \mu_i$ for each $i \in V$.

We first describe Tighten-Label$(f, \mu, 0)$, when it is essentially a multiplicative interpretation if 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 $V\setminus V'$, 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 μ 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 $V'\setminus S \ne \varnothing$, let us multiply μ_i for each $i \in V'\setminus S$ by α defined as

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

By the definition of S, $\alpha > 1$, and after multiplying by α, at least one arc $ij \in E_f$ with $i \in V'\setminus S$, $j \in S$ will become tight. Tight arcs inside S also remain tight, hence S is extended by at least one node. Also, the choice of α guarantees that μ remains conservative. Note that arcs with $j \in V\setminus S$ may disappear from $E^{\mu}_f(\Delta)$ as their fatness decreases.

Let us now describe Tighten-Label(f, μ, Δ) for $\Delta > 0$. The main difference is that increasing μ_i may turn a Δ-positive node into Δ-neutral. We have to stop increasing μ_i at this point and add i to S. (This is in accordance with the goal that we allow tight paths to Δ-neutral nodes as well.) In each phase of the algorithm, let $S \subseteq V$ denote the subset of nodes that have a (possibly empty) tight path in $E^{\mu}_f(\Delta)$ to a Δ-negative or Δ-neutral node. S is initialized as the set of Δ-negative and Δ-neutral nodes and is extended by at least one element per phase. The algorithm terminates once $S = V$.

In every phase, we multiply μ_i for every $i \in V\setminus S$ by the same factor $\alpha > 1$. Consider an arc $ij \in E_f(\Delta)$ with $i \in V\setminus S$, $j \in S$. This must satisfy $\gamma^{\mu}_{ij} < 1$. Increasing μ_i increases γ^{μ}_{ij}. Note that the fatness $s^{\mu}_f(ij) = s_f(ij)/\mu_j$ is not changed as it is not dependent of μ_i. By definition, all nodes in $V\setminus S$ are Δ-positive. When increasing μ_i, e^{μ}_i decreases and therefore i may become Δ-neutral. At this point, we have to stop to avoid creating new Δ-negative nodes. Let us define

$$\alpha := \min\left\{\min\left\{\frac{1}{\gamma^{\mu}_{ij}} \colon ij \in E^{\mu}_f(\Delta),\ i \in V\setminus S, j \in S\right\}, \min\left\{\frac{e^{\mu}_i}{d_i\Delta} \colon i \in V\setminus S\right\}\right\}. \tag{4}$$

Algorithm Symmetric Fat-Path
for $i \in V$ **do** $\mu_i \leftarrow 1/M_i$;
for $ij \in E$ **do** $f_{ij} \leftarrow 0$;
$\Delta \leftarrow MB^2 + 1$;
while $(2n+6m)\Delta \geq 1/B^m$ **do**
  do
    Tighten-Label(f, μ, Δ);
    $D \leftarrow \{i \in V: e_i^\mu > (d_i + 1)\Delta\}$;
    $N_0 \leftarrow \{i \in V: e_i^\mu \leq d_i\Delta\}$;
    pick $s \in D$, $t \in N_0$ connected by a tight path P;
    send Δ units of flow along P;
  while $D \neq \varnothing$;
  Adjust$(\Delta, \Delta/2)$;
  $\Delta \leftarrow \Delta/2$;
Adjust$(\Delta, 0)$;
Tighten-Label$(f, \mu, 0)$;
compute a maximum flow from nodes $\{s \in V: e_s > 0\}$ to nodes $\{t \in V: e_t < 0\}$ using tight arcs for μ;
return optimal primal solution f and optimal dual solution μ.

FIGURE 1. The algorithm for symmetric linear generalized flows.

Clearly, $\alpha > 1$, and after multiplying each μ_i by α for $i \in V \backslash S$, either we obtain a new Δ-neutral node in $V \backslash S$, or at least one arc $ij \in E_f$ with $i \in V \backslash S$, $j \in S$ will become tight. Tight arcs inside S also remain tight, and their capacities are unchanged, hence S is extended by at least one node. Also, the choice of α guarantees that μ remains Δ-conservative.

Note that in Tighten-Label$(f, \mu, 0)$ we first identified the set $V' \subseteq V$, and set $\mu_i = \infty$ for $i \in V \backslash V'$. In contrast, for $\Delta > 0$ we skip this step and increase μ_i on all nodes of $V \backslash S$. The reason for this is that for $\Delta > 0$, every $i \in V \backslash S$ becomes Δ-neutral if μ_i is sufficiently increased; the second term in the definition (4) of α accounts for this effect. Therefore no μ_i values will be set to ∞.

Further, if $\Delta > 0$, at the termination of Tighten-Label(f, μ, Δ), there must exist some Δ-negative or Δ-neutral nodes. Indeed, if we start with $S = \varnothing$, then some nodes must become Δ-neutral in the first phase.

5.4. Description of the algorithm. The algorithm is shown in Figure 1. We start with $\mu_i = 1/M_i$ for every node $i \in V$, and $f \equiv 0$. The algorithm consists of Δ-phases, and Δ decreases by a factor of 2 between two phases. The initial value of Δ is $MB^2 + 1$. Once $(2n+6m)\Delta < 1/B^m$, then an optimal solution can be found by a single maximum flow computation, as shown in §5.6. In this case we terminate.

During the Δ-phase, we maintain a pseudoflow f along with a Δ-conservative labeling μ. The μ_i values can only increase. Let N_0 denote the set of Δ-negative and Δ-neutral nodes, and D the set of nodes i with $e_i^\mu > (d_i + 1)\Delta$. The Δ-phase consists of a sequence of iterations until D becomes empty. In every iteration of the algorithm, we update μ to a Δ-canonical labeling by calling Tighten-Label(f, μ, Δ). Note that after Tighten-Label(f, μ, Δ), we have $N_0 \neq \varnothing$ as remarked above; further, there exists a tight path from every node in $v \in V$ to a node in N_0. If $D \neq \varnothing$ still holds, we pick an arbitrary $s \in D$, and send Δ units of flow from s to some Δ-negative or Δ-neutral $t \in N_0$ on a tight path P. At the end of the Δ-phase, we modify f by Adjust$(\Delta, \Delta/2)$, and proceed to the $\Delta/2$-phase.

5.5. Analysis.

CLAIM 1. *The initial μ is Δ-conservative, and Δ-conservativeness is maintained during the entire Δ-phase.*

PROOF. At the beginning $s_f^\mu(ij) = \gamma_{ij} u_{ij}/\mu_j \leq MB^2 < \Delta$ for every $ij \in E$, and E_f contains no backward arcs. Consequently, $E_f^\mu(\Delta) = \varnothing$ and Δ-conservativeness trivially holds. Also, $\mu_i = 1/M_i$ holds for every node i. Tighten-Label(f, μ, Δ) clearly maintains Δ-conservativeness. This is also maintained when sending flow, as we only use tight arcs. At the end of the Δ-phase, Adjust$(\Delta, \Delta/2)$ transforms f to a $\Delta/2$-conservative pseudoflow. $\square$

CLAIM 2. *At the beginning of every Δ-phase, $Ex_\Delta^\mu(f) \leq (2n+3m)\Delta$.*

PROOF. In the first phase, $Ex_\Delta^\mu(f) \leq M(\sum_{i \in V} b_i) \leq MBn < (2n+3m)\Delta$. Once we finish all iterations in the Δ-phase, $D = \varnothing$ implies $Ex_\Delta^\mu(f) \leq n\Delta$. In Adjust$(\Delta, \Delta/2)$, we increase the excess by at most $3m\Delta/2$ (Lemma 2.) Hence at the beginning of the $\Delta/2$-phase, $Ex_{\Delta/2}^\mu(f) \leq (2n+3m)\Delta/2$, proving the claim. $\square$

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

LEMMA 3. *A Δ-phase consists of at most $2n+3m$ iterations.*

PROOF. Let $\Psi(i) := \lfloor e_i^\mu/\Delta \rfloor - d_i$ if $e_i^\mu > (d_i+1)\Delta$ and $\Psi(i) := 0$ otherwise. Consider the potential function $\Psi := \sum_{i\in V} \Psi(i)$. Clearly, $\Psi \le Ex_\Delta^\mu(f)/\Delta$ and therefore $\Psi \le 2n+3m$ holds at the beginning by Claim 2. In the relabeling steps, Ψ may only decrease, and in every path augmentation, it decreases by exactly 1. □

THEOREM 3. *The algorithm runs in $O(m(m+n\log n)(m\log B+\log M))$ time.*

PROOF. The value of Δ always decreases by a factor of 2, its initial value is MB^2, and we terminate if $Ex_\Delta^\mu(f) < 1/B^m$. Hence the number of phases can be bounded by $O(m\log B+\log M)$. The number of iterations in one phase is $O(m)$ by Lemma 3. The running time of an iteration is dominated by the Tighten-Label step, that can be done in $O(m+n\log n)$ time following Fredman and Tarjan's [11] implementation of Dijkstra's algorithm. The step Adjust$(\Delta, \Delta/2)$ can be performed in $O(m)$ time. At termination, we need one further maximum flow computation as described in the next section; it can be done in running time $O(n^3)$ (Ahuja et al. [2, Chapter 7]). □

5.6. Moving to an optimal solution. If $(2n+6m)\Delta < 1/B^m$ at a certain iteration of the algorithm, then by Claim 2, $Ex_\Delta^\mu(f)+3m\Delta < 1/B^m$ holds. Next, we transform f to $\bar f$ by Adjust$(\Delta, 0)$, so that μ is a conservative labeling for $\bar f$. By Lemma 2, $Ex^\mu(\bar f) < 1/B^m$ follows. Then Tighten-Label$(\bar f, \mu, 0)$ transforms μ into a canonical labeling. As shown below, a single maximum flow computation yields an optimal solution. This is the standard technique of how most algorithms in the literature terminate.

Given a canonical labeling μ for f, let $P := \{i\in V\colon e_i > 0, \mu_i < \infty\}$ and $N := \{i \in V\colon e_i < 0\}$. Let us construct the following maximum flow instance. Let $\tilde V := V \cup \{s,t\}$, and let $\tilde E$ consist of all tight arcs in E_f, plus an si arc for every $i\in P$, and a ti arc for every $i \in N$. Let us set the capacity equal to the relabeled residual capacity on the tight arcs in E_f; for every arc si, let us set the capacity equal to e_i^μ and for every arc it, equal to $-e_i^\mu$. Let us compute a maximum flow g from s to t in this network, and modify f_{ij} by $g_{ij}\mu_i$ on every $ij \in E_f$. Let f' denote the resulting pseudoflow.

LEMMA 4. *The resulting pseudoflow f' is an optimal solution.*

PROOF. Since flow was sent only on tight arcs, μ is also conservative for f'. If there are no more nodes i with $e_i^\mu(f') > 0$, then by Theorem 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. If an arc $ij \in E$ leaves S then either ij is saturated, that is, $f'_{ij} = u_{ij}$, or $\gamma_{ij}^\mu < 1$ and $f'_{ij} = 0$. Let F_0 denote the set of these saturated arcs. Similarly, if $ij\in E$ enters S, then $f'_{ij} = 0$ must hold. Also, on all arcs ij with $i, j \in S$, $f'_{ij} > 0$ either $\gamma_{ij}^\mu = 1$ or $f'_{ij} = u_{ij}$. Let F_1 denote the set of such arcs with $f'_{ij} < u_{ij}$ (and $\gamma_{ij}^\mu = 1$), and F_2 the set of those with $f'_{ij} = u_{ij}$. Let $e_i^\mu(f')$ denote the excess with respect to the flow f'. Therefore,

$$\begin{aligned}
0 \le Ex^\mu(f') &= \sum_{i\in S} e_i^\mu(f') = \sum_{i\in S}\left(\sum_{j:\, ji\in E} \gamma_{ji}^\mu f'^{\mu}_{ji} - \sum_{j:\, ij\in E} f'^{\mu}_{ij} - b_i^\mu\right)\\
&= \sum_{ij\in F_1} (\gamma_{ji}^\mu f'^{\mu}_{ji} - f'^{\mu}_{ji}) + \sum_{ij\in F_2} u_{ij}^\mu(\gamma_{ij}^\mu - 1) - \sum_{ij\in F_0} u_{ij}^\mu - \sum_{i\in S} b_i^\mu\\
&= \sum_{ij\in F_2} u_{ij}\left(\frac{\gamma_{ij}}{\mu_j} - \frac{1}{\mu_i}\right) - \sum_{ij\in F_0}\frac{u_{ij}}{\mu_i} - \sum_{i\in S}\frac{b_i}{\mu_i}.
\end{aligned} \tag{5}$$

Let $B^* \le B^m$ denote the product of the denominators of the γ_{ij}'s. We claim that every term in the above expression is an integer multiple of $1/B^*$. Indeed, using that μ is a canonical labeling for f, there exists a tight path P_i from each node i to a negative node t. Then $1/\mu_i = M_t\gamma(P_i)$ and is hence an integer multiple of $1/B^*$. Similarly, every γ_{ij}/μ_j is also an integer multiple of $1/B^*$. For this, note that since $\gamma_{ij}^\mu \ne 1$, the arc ij cannot be contained on the tight path P_i. Since the u_{ij}'s and b_i's are integers, this verifies the claim. By the assumption of the lemma, $0 \le Ex^\mu(f') \le Ex^\mu(f) < 1/B^m \le 1/B^*$. This implies $Ex^\mu(f') = 0$, that is, $\mu_i = \infty$ whenever $e_i > 0$. (Note that the subroutine Tighten-Label$(\bar f, \mu, 0)$ might set certain labels to ∞.) By Theorem 2, this shows optimality. □

6. Concave generalized flows algorithm. We describe the algorithm in the same structure as for the linear case: §6.1 presents the optimality conditions; Δ-conservative and Δ-canonical labels are discussed in §§6.2 and 6.3, respectively. Section 6.4 presents the algorithm, and §6.5 gives its analysis.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

6.1. Optimality conditions. The characterization of optimality was given in Shigeno [36]; we have to modify the results slightly as we use the symmetric formulation. Let us call an arc $ij \in E$ *immense*, if $\Gamma_{ij}(l_{ij}) = -\infty$, and other arcs *regular*. First, let us transform the problem to an equivalent instance with (i) $l_{ij} = 0$ for every arc and $\Gamma_{ij}(0) = 0$ for every regular arc; and (ii) every gain function Γ_{ij} is strictly monotone increasing on $[0, u_{ij}]$.

For (i), on each arc $ij \in E$, let us replace u_{ij} by $u_{ij} - l_{ij}$ and l_{ij} by 0. If ij is a regular arc, we modify the gain function to $\Gamma_{ij}(\alpha + l_{ij}) - \Gamma_{ij}(l_{ij})$, and if ij is an immense arc, to $\Gamma_{ij}(\alpha + l_{ij})$. Accordingly for every $i \in V$, let us increase b_i by $\sum_{j:\, ij\in E} l_{ij}$, and decrease it by the sum of the $\Gamma_{ji}(l_{ji})$'s on regular arcs.

For (ii), let us define $\tilde{u}_{ij} = \inf\{p\colon 0 \le p \le u_{ij}, \Gamma_{ij}(p) = \Gamma_{ij}(u_{ij})\}$. By continuity, $\Gamma_{ij}(\tilde{u}_{ij}) = \Gamma_{ij}(u_{ij})$, and $\Gamma_{ij}(u_{ij})$ is strictly monotone increasing on the interval $[0, \tilde{u}_{ij}]$. Let us replace u_{ij} by $\tilde{u}_{ij} \le u_{ij}$.

For a pseudoflow $f\colon E \to \mathbb{R}$, we define the residual network $G_f = (V, E_f)$ identical as for the generalized flow setting: $ij \in E_f$ if $ij \in E$ and $f_{ij} < u_{ij}$ 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 $\Gamma_{ji}(\alpha)\colon [-\Gamma_{ij}(u_{ij}), -\Gamma_{ij}(0)] \to [-u_{ij}, 0]$ by

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

Hence $\Gamma_{ji}(f_{ji}) = -f_{ij}$.

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

$$\frac{\Gamma_{ij}(\alpha + \Delta') - \Gamma_{ij}(\alpha)}{\Delta'} \le \frac{\Gamma_{ij}(\alpha + \Delta) - \Gamma_{ij}(\alpha)}{\Delta} \le \Gamma_{ij}^{+}(\alpha), \tag{6a}$$

$$\frac{\Gamma_{ij}(\alpha) - \Gamma_{ij}(\alpha - \Delta')}{\Delta'} \ge \frac{\Gamma_{ij}(\alpha) - \Gamma_{ij}(\alpha - \Delta)}{\Delta} \ge \Gamma_{ij}^{-}(\alpha) \tag{6b}$$

for $0 \le \alpha \le u_{ij} - \Delta'$ and for $\Delta' \le \alpha \le u_{ij}$, respectively. Furthermore, if $0 < \alpha < \alpha' < u_{ij}$, then $\Gamma_{ij}^{+}(\alpha') \le \Gamma_{ij}^{-}(\alpha') \le \Gamma_{ij}^{+}(\alpha) \le \Gamma_{ij}^{-}(\alpha)$. The following claim is easy to verify.

Claim 3. *For any $ij \in E$ with $0 < f_{ij} < u_{ij}$, $\Gamma_{ij}^{+}(f_{ij}) = 1/\Gamma_{ji}^{-}(f_{ji})$, $\Gamma_{ij}^{-}(f_{ij}) = 1/\Gamma_{ji}^{+}(f_{ji})$.* □

Let $P = i_0, \ldots, i_k$ be a walk in the auxiliary graph E_f. By sending α units of flow along P, we mean the following. First we increase $f_{i_0 i_1}$ by α 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 $h = 1, \ldots, k-1$, we increase the flow on $i_h i_{h+1}$ by β and set the new value of β 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 α is chosen small enough so that no capacity gets violated. Let $f^{\alpha, P}$ denote the modified flow.

If $C = i_0, \ldots, i_{k-1}$ is a cycle in E_f, then by sending α units of flow around C from i_0 we mean sending α units on the walk $i_0, \ldots, i_{k-1} i_0$. This modifies e_i only in node i_0: if the flow increase from $i_{k-1} i_0$ is bigger than α, then e_{i_0} increases, and if it is smaller then it decreases. The next lemma characterizes when e_{i_0} can increase. For an arbitrary walk P in E_f, let $\Gamma_f^{+}(P) = \prod_{e\in P} \Gamma_e^{+}(f_e)$.

Lemma 5. *Let C be a cycle in E_f with $i \in V(C)$. If $\Gamma_f^{+}(C) > 1$ then e_i can be increased by sending some flow around C. If $\Gamma_f^{+}(C) \le 1$, then it is not possible to increase e_i by sending any amount of flow around C.*

Since this property is independent from the choice of i, 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 4. *Let $P = i_0 i_1, \ldots, i_k$ be a walk in E_f. 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 $P_h = i_0, \ldots, i_h$ for $h = 0, \ldots, k$. There is nothing to prove for $h = 0$; assume we have already proved it for P_{h-1}. We want to find a $\delta > 0$ for some $\varepsilon > 0$ satisfying the claim for P_h. First, it is possible to pick a small enough $\varepsilon^* > 0$ such that

$$\Gamma_f^{+}(P_h) - \varepsilon < (\Gamma_{i_{h-1} i_h}^{+}(f_{i_{h-1} i_h}) - \varepsilon^*)(\Gamma_f^{+}(P_{h-1}) - \varepsilon^*). \tag{7}$$

By the definition of $\Gamma_{i_{h-1} i_h}^{+}$, there exists a $\delta^* > 0$ such that for any $0 < \beta \le \delta^*$,

$$(\Gamma_{i_{h-1} i_h}^{+}(f_{i_{h-1} i_h}) - \varepsilon^*)\beta \le \Gamma_{i_{h-1} i_h}(f_{i_{h-1} i_h} + \beta) - \Gamma_{i_{h-1} i_h}(f_{i_{h-1} i_h}). \tag{8}$$

By induction for P_{h-1} and ε^*, we can choose a small enough $\delta > 0$ such that we have the following properties for every $0 < \alpha \le \delta$: $\Gamma_f^{+}(P_{h-1})\alpha \le \delta^*$, and the increase of e_{h-1} for $f^{\alpha, P_{h-1}}$ is at least $\beta = (\Gamma_f^{+}(P_{h-1}) - \varepsilon^*)\alpha$. Then (7) and (8) show that for $0 < \alpha \le \delta$, $f^{\alpha, P}$ increases e_{i_h} by at least $(\Gamma_f^{+}(P) - \varepsilon)\alpha$. □

The definition of GAPs is identical to the linear case, with the only difference that $\Gamma^+_f(P) > M_s/M_t$ instead of $\gamma(P) > M_s/M_t$ in case (c).

The following lemma can be proved similarly as Lemma 1.

LEMMA 6. *If f is an optimal solution, then no GAP may exist.* □

Relabelings are also defined analogously as for generalized flows. Given μ: $V \to \mathbb{R}_{>0} \cup \{\infty\}$, 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, excesses, and capacities are $b^\mu_i := b_i/\mu_i$, $e^\mu_i := e_i/\mu_i$, and $u^\mu_{ij} := u_{ij}/\mu_i$. A relabeling is *conservative*, if for any residual arc $ij \in E_f$, $\Gamma^{\mu+}_{ij}(f^\mu_{ij}) \le 1$, that is, no edge may increase the relabeled flow. Furthermore we require $\mu_i \ge 1/M_i$ for every $i \in V$ with equality whenever $e_i < 0$.

We use the same convention for infinite μ_i values as for generalized flows. If $\mu_i = \infty$, we define $b^\mu_i = e^\mu_i = 0$, $u^\mu_{ij} = 0$ for $ij \in E$, and furthermore $\Gamma^{\mu+}_{ji}(f^\mu_{ji}) = 0$ for all arcs $ji \in E_f$. Finally, for $ij \in E_f$ with $\mu_i = \infty$, $\mu_j < \infty$, let $\Gamma^{\mu+}_{ij}(f^\mu_{ij}) = \infty$.

If μ 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 5. $\Gamma^{\mu+}_{ij}(\alpha) = (\mu_i/\mu_j)\Gamma^+_{ij}(\alpha)$, *and* $\Gamma^{\mu-}_{ij}(\alpha) = (\mu_i/\mu_j)\Gamma^-_{ij}(\alpha)$. □

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

THEOREM 4 (SHIGENO [36]). *For a pseudoflow f, 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 μ 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 μ_i being the reciprocal of the Lagrange multiplier corresponding to $e_i + \kappa_i \ge 0$. (i) implies (ii) by Lemma 6. The proof of (ii)⇒(iii) is the same as in Theorem 2, with $\gamma(P)$ replaced by $\Gamma^+_f(P)$. □

6.2. Δ-conservative labelings. We define the notion of Δ-conservative labeling analogously as in §5.3 for the linear case. Let us define the *fatness* of $ij \in E_f$ by $s_f(ij) := \Gamma_{ij}(u_{ij}) - \Gamma_{ij}(f_{ij})$ (if ij is a backward arc, this is equivalent to $s_f(ij) = f_{ji}$). The fatness expresses the maximum possible flow increase in j if we saturate ij. This notion enables us to identify arcs that can participite in fat paths during the algorithm. In accordance with the other variables, the relabeled fatness is defined as $s^\mu_f(ij) := s_f(ij)/\mu_j$.

Consider a scaling parameter $\Delta > 0$. As in the linear case, the Δ-*fat graph* $E^\mu_f(\Delta)$ is the set of residual arcs of relabeled fatness at least Δ:

$$E^\mu_f(\Delta) = \{ij \in E_f\colon s^\mu_f(ij) \ge \Delta\}.$$

Arcs in $E^\mu_f(\Delta)$ will be called Δ-*fat arcs*. As in Shigeno [36], we use the following linearization on Δ-fat arcs in "chunks" of Δ:

$$\theta^\mu_\Delta(ij) := \frac{\Delta\mu_i}{\Gamma^{-1}_{ij}(\Gamma_{ij}(f_{ij}) + \Delta\mu_j) - f_{ij}}, \quad ij \in E^\mu_f(\Delta). \tag{9}$$

This is well defined since $\Gamma_{ij}(f_{ij}) + \Delta\mu_j \le \Gamma_{ij}(u_{ij})$ for Δ-fat arcs. Note that if $\Gamma_{ij}(\cdot)$ is linear, i.e., $\Gamma_{ij}(\alpha) = \gamma_{ij}\alpha$, then $\theta^\mu_\Delta(ij) = \gamma^\mu_{ij}$. An equivalent way of writing (9) is

$$\theta^\mu_\Delta(ij) = \frac{\Delta}{\Gamma^{\mu-1}_{ij}(\Gamma^\mu_{ij}(f^\mu_{ij}) + \Delta) - f^\mu_{ij}}, \quad ij \in E^\mu_f(\Delta). \tag{10}$$

Also, if the reverse arc ji is Δ-fat, then using $\Gamma_{ji}(f_{ji}) = -f_{ij}$ and $\Gamma^{-1}_{ji} = -\Gamma_{ij}(-\alpha)$, we get

$$\theta^\mu_\Delta(ji) = \frac{\Delta\mu_j}{\Gamma_{ij}(f_{ij}) - \Gamma_{ij}(f_{ij} - \Delta\mu_i)}. \tag{11}$$

Consider a label function μ: $V \to \mathbb{R}_{>0} \cup \{\infty\}$. A node $i \in V$ is called Δ-*negative* if $e^\mu_i < d_i\Delta$, Δ-*neutral* if $e^\mu_i = d_i\Delta$, and Δ-*positive* if $e^\mu_i > d_i\Delta$. The labeling μ is Δ-*conservative*, if $\theta^\mu_\Delta(ij) \le 1$ holds for every $ij \in E^\mu_f(\Delta)$. Furthermore, we require $\mu_i \ge 1/M_i$ for all $i \in V$, with equality for every Δ-negative node i.

Note that a Δ-conservative labeling cannot have any nodes with $\mu_i = \infty$. Using the convexity of Γ^{-1}, it can be shown that if $\Delta' \ge \Delta$ then $\theta^\mu_{\Delta'}(ij) \le \theta^\mu_\Delta(ij)$ for every arc $ij \in E^\mu_f(\Delta')$. Let $Ex^\mu(f) = \sum_{i\in V} \max\{e^\mu_i, 0\}$ and $Ex^\mu_\Delta(f) = \sum_{i\in V} \max\{e^\mu_i - d_i\Delta, 0\}$ denote the total relabeled excess of positive and Δ-positive nodes, respectively.

The key importance of Δ-conservativeness is that it is maintained when sending Δ units of flow on arcs with $\theta^\mu_\Delta(ij) = 1$. This is formulated in the next simple lemma.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

LEMMA 7. *Assume μ is Δ-conservative, and let $ij \in E_f^\mu(\Delta)$ be an arc with $\theta_\Delta^\mu(ij) = 1$. If we increase f_{ij}^μ by Δ, then $\Gamma_{ij}^\mu(f_{ij}^\mu)$ also increases by Δ, and Δ-conservativeness is maintained.*

PROOF. The condition $\theta_\Delta^\mu(ij) = 1$ is equivalent to $\Gamma_{ij}^\mu(f_{ij}^\mu + \Delta) = \Gamma_{ij}^\mu(f_{ij}^\mu) + \Delta$, showing the first part. Let $\bar{f}_{ij} = f_{ij} + \Delta\mu_i$ be the modified flow. For the second part, $\theta_\Delta^\mu(ij) \le 1$ for $\bar{f}_{ij}$ easily follows from convexity. Further, observe (11) shows that we get $\theta_\Delta^\mu(ji) = 1$ for $\bar{f}_{ij}$. This gives Δ-conservativeness for the modified flow as all other arcs are left unchanged. □

In contrast to Lemma 2, the following claim only enables to transform a Δ-conservative labeling a $\Delta/2$-conservative one, instead of arbitrary $\Delta' < \Delta$. The reason is that besides the set of Δ'-fat arcs being larger than the Δ-fat arcs, we may also have Δ-fat arcs with $\theta_\Delta^\mu(ij) \le 1 < \theta_{\Delta'}^\mu(ij)$.

LEMMA 8. *Let f be a pseudoflow with a Δ-conservative labeling μ. Then there exists a flow $\bar{f}$ such that μ is $\Delta/2$-conservative for $\bar{f}$ and $Ex_{\Delta/2}^\mu(\bar{f}) \le Ex_\Delta^\mu(f) + \frac{3}{2}m\Delta$.*

PROOF. Consider a $\Delta/2$-fat arc ij with $\theta_{\Delta/2}^\mu(ij) > 1$ for f, that is,

$$\Gamma_{ij}^{-1}\left(\Gamma_{ij}(f_{ij}) + \frac{\Delta}{2}\mu_j\right) - f_{ij} < \frac{\Delta}{2}\mu_i. \tag{12}$$

There are two possible scenarios: (a) ij was not Δ-fat, that is,

$$\frac{\Delta}{2}\mu_j \le \Gamma_{ij}(u_{ij}) - \Gamma_{ij}(f_{ij}) < \Delta\mu_j, \tag{13}$$

or (b) ij was also a Δ-fat arc. Then by Δ-conservativeness,

$$\Gamma_{ij}^{-1}(\Gamma_{ij}(f_{ij}) + \Delta\mu_j) - f_{ij} \ge \Delta\mu_i. \tag{14}$$

In both cases, let us define

$$\bar{f}_{ij} = \Gamma_{ij}^{-1}\left(\Gamma_{ij}(f_{ij}) + \frac{\Delta}{2}\mu_j\right).$$

The $\Delta/2$-fatness of ij guarantees that this is well defined. In case (a), we claim that ij is not $\Delta/2$-fat for $\bar{f}$. Indeed,

$$\Gamma_{ij}(u_{ij}) - \Gamma_{ij}(\bar{f}_{ij}) = \Gamma_{ij}(u_{ij}) - \left(\Gamma_{ij}(f_{ij}) + \frac{\Delta}{2}\mu_j\right) < \frac{\Delta}{2}\mu_j.$$

The last inequality follows by the second part of (13). In case (b), we claim that if ij is a $\Delta/2$-fat arc for $\bar{f}$ then $\theta_{\Delta/2}^\mu(ij) \le 1$ must hold for $\bar{f}$. Indeed, if we subtract (12) from (14), we get

$$\Gamma_{ij}^{-1}(\Gamma_{ij}(f_{ij}) + \Delta\mu_j) - \Gamma_{ij}^{-1}\left(\Gamma_{ij}(f_{ij}) + \frac{\Delta}{2}\mu_j\right) > \frac{\Delta}{2}\mu_i,$$

and by substituting $\bar{f}_{ij}$, it follows that

$$\Gamma_{ij}^{-1}\left(\Gamma_{ij}(\bar{f}_{ij}) + \frac{\Delta}{2}\mu_j\right) - \bar{f}_{ij} > \frac{\Delta}{2}\mu_i,$$

that is, $\theta_{\Delta/2}^\mu(ij) < 1$ for $\bar{f}$.

We next show that if ji is also a $\Delta/2$-fat arc for $\bar{f}$, then $\theta_{\Delta/2}^\mu(ji) \le 1$ holds for $\bar{f}$. Indeed, using (11), $\theta_{\Delta/2}^\mu(ji) \le 1$ for $\bar{f}$ is equivalent to

$$\Gamma_{ij}(\bar{f}_{ij}) - \Gamma_{ij}\left(\bar{f}_{ij} - \frac{\Delta}{2}\mu_i\right) \ge \frac{\Delta}{2}\mu_j.$$

Equivalently,

$$\Gamma_{ij}(f_{ij}) + \frac{\Delta}{2}\mu_j - \Gamma_{ij}\left(\bar{f}_{ij} - \frac{\Delta}{2}\mu_i\right) \ge \frac{\Delta}{2}\mu_j.$$

Subtracting $(\Delta/2)\mu_j$, rearranging and applying the strictly monotone increasing function Γ_{ij}^{-1}, we get $f_{ij} \ge \bar{f}_{ij} - (\Delta/2)\mu_i$, that follows from (12) by substituting $\bar{f}_{ij}$.

We define $\bar{f}_{ij}$ the above way whenever ij is a $\Delta/2$-fat arc with $\theta_\mu(ij) > 1$. (As a simple consequence of concavity, this cannot be the case for both ij and ji.) If this does not hold for neither ij nor ji, then let $\bar{f}_{ij} = f_{ij}$. The next claim compares f_{ij} to $\bar{f}_{ij}$ and $\Gamma(f_{ij})$ to $\Gamma(\bar{f}_{ij})$.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

Algorithm Symmetric Concave Fat-Path
for $i \in V$ **do** $\mu_i \leftarrow 1/M_i$;
for $ij \in E$ **do** $f_{ij} \leftarrow u_{ij}$;
$\Delta \leftarrow MU + 1$;
while $(2n+4m)\Delta \geq \varepsilon$ **do**
  do
    Tighten-Label(f, μ, Δ);
    $D \leftarrow \{i \in V: e_i^\mu > (d_i+1)\Delta\}$;
    $N_0 \leftarrow \{i \in V: e_i^\mu \leq d_i\Delta\}$;
    pick $s \in D$, $t \in N_0$ connected by a tight path P;
    send Δ units of flow along P;
  while $D \neq \varnothing$;
  Adjust(Δ);
  $\Delta \leftarrow \Delta/2$;
return ε-approximate optimal solution f.

FIGURE 2. The algorithm for symmetric concave generalized flows.

CLAIM 6. $|\bar{f}_{ij}^\mu - f_{ij}^\mu| \leq \Delta/2$ *and* $|\Gamma_{ij}^\mu(\bar{f}_{ij}^\mu) - \Gamma_{ij}^\mu(f_{ij}^\mu)| \leq \Delta/2$.

PROOF. There is nothing to prove if $\bar{f}_{ij} = f_{ij}$. Assume f_{ij} was increased as above (decreasing f_{ij} is the same as increasing f_{ji}). The first part follows by (12). By the definition of $\bar{f}_{ij}$,

$$\Gamma_{ij}(\bar{f}_{ij}) - \Gamma_{ij}(f_{ij}) = \Gamma_{ij}(f_{ij}) + \frac{\Delta}{2}\mu_j - \Gamma_{ij}(f_{ij}) = \frac{\Delta}{2}\mu_j,$$

giving the second part. □

For $\Delta/2$-conservativeness, we also need to show that $\bar{f}$ has no $\Delta/2$-negative nodes with $\mu_i > 1/M_i$. By the above claim, the total possible change of relabeled flow on arcs incident to i is $d_i\Delta/2$. A node is nonnegative for Δ if $e_i^\mu \geq d_i\Delta$ and for $\Delta/2$ if $e_i^\mu \geq d_i\Delta/2$. Consequently, a Δ-nonnegative node cannot become $\Delta/2$-negative.

Finally, $Ex_{\Delta/2}^\mu(f) \leq Ex_\Delta^\mu(f) + \sum_{i\in V} d_i\Delta/2$, and each arc is responsible for creating at most $\Delta/2$ units of new excess. This gives $Ex_{\Delta/2}^\mu(\bar{f}) \leq Ex_\Delta^\mu(f) + (3m/2)\Delta$, as required. □

The subroutine Adjust(Δ) performs the simple modifications described in the proof (in contrast to the linear case, this subroutine has only one parameter).

6.3. Δ-canonical labelings. Given a pseudoflow f and a Δ-conservative labeling μ, the arc $ij \in E_f^\mu(\Delta)$ is called *tight* if $\theta_\Delta^\mu(ij) = 1$. A directed path in $E_f^\mu(\Delta)$ is called tight if it consists of tight arcs. We call μ a Δ-*canonical* labeling, if from each node i there exists a tight path to a Δ-negative or to a Δ-neutral node. Such a path is approximately a highest gain Δ-fat augmenting path. The subroutine Tighten-Label(f, μ, Δ) returns a Δ-canonical label $\mu' \geq \mu$ for a Δ-conservative label μ. This is almost identical to the algorithm described in §5.3. The only difference is in the definition of the multiplier α, which is given by (4) for the linear case. Instead, we define

$$\alpha := \min\left\{\min\left\{\frac{1}{\theta_\Delta^\mu(ij)}: ij \in E_f^\mu(\Delta),\ i \in V\setminus S,\ j \in S\right\}, \min\left\{\frac{e_i^\mu}{d_i\Delta}: i \in V\setminus S\right\}\right\}.$$

In an iteration, we multiply every μ_i by α for $i \in V\setminus S$, where S is the set of nodes from which there exists a tight path to a Δ-negative or a Δ-neutral node. We claim that as in the linear case, this maintains Δ-conservativeness, and extends S by at least one node. This is a simple consequence of the fact that multiplying μ_i by α multiplies $\theta_\Delta^\mu(ij)$ by α for every incident arc ij.

To verify that μ remains Δ-conservative, we also have to check $\theta_\Delta^\mu(ij) \leq 1$ on all arcs $ij \in E_f^\mu(\Delta)$, $j \in V\setminus S$. This follows by the convexity of Γ_{ij}^{-1}.

As in the linear case, the set of Δ-negative and Δ-neutral nodes will be nonempty after performing Tighten-Label(f, μ, Δ), with every node $v \in V$ connected by a tight path to such a node. (Note that $\Delta > 0$ is always assumed in the nonlinear case.)

6.4. The main algorithm. The algorithm is shown on Figure 2. Let us initialize $\mu_i = 1/M_i$ for every $i \in V$, and $f_{ij} = u_{ij}$ for every $ij \in E$. (We set f_{ij} to the upper bounds rather than the lower bounds because $\Gamma_{ij}(0) = -\infty$ is allowed.) The algorithm consists of Δ-phases, and Δ decreases by a factor of 2 between two phases. The initial value of Δ is $\Delta = MU + 1$. The algorithm terminates with an ε-approximate solution once $(2n+4m)\Delta < \varepsilon$.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

During the Δ-phase, we maintain a pseudoflow f and a Δ-conservative labeling μ. The μ_i values may only increase. Let D denote the set of nodes i with $e_i^\mu > (d_i+1)\Delta$ and let N_0 denote the set of Δ-negative or Δ-neutral nodes. The Δ-phase consists of iterations, and terminates whenever D becomes empty. In each iteration, we update μ to a canonical labeling by calling Tighten-Label(f, μ, Δ). If $D \neq \varnothing$ still holds, then we send Δ units of relabeled flow on a tight path from some $s \in D$ to a Δ-neutral or Δ-negative node t. (Note that $N_0 \neq \varnothing$ always holds after performing Tighten-Label(f, μ, Δ), and every node is connected by a tight path to N_0.)

6.5. Analysis.

Claim 7. *The initial μ is Δ-conservative, and Δ-conservativeness is maintained during the entire Δ-phase.*

Proof. Initially, $f \equiv u$ and hence E_f is the set of backward arcs. For an arc $ij \in E$, $s_\Delta^\mu(ji) = u_{ij}/\mu_i \leq MU < \Delta$, and hence $E_f^\mu(\Delta) = \varnothing$. Also, $\mu_i = 1/M_i$ holds for every node i. Tighten-Label(f, μ, Δ) clearly maintains Δ-conservativeness. We use only tight arcs to send flow, and Lemma 7 guarantees that this preserves Δ-conservativeness. At the end of the Δ-phase, Adjust(Δ) transforms f to a $\Delta/2$-conservative pseudoflow. □

Claim 8. *The Δ-phase starts with $Ex_\Delta^\mu(f) \leq (2n+3m)\Delta$.*

Proof. For the initial solution, $Ex_\Delta^\mu(f) \leq M(\sum_{i\in V}|b_i| + mU) \leq (m+n)MU \leq (m+n)\Delta$, since $\Delta = MU+1$. Once we finish all iterations in the Δ-phase, $D = \varnothing$ implies $Ex_\Delta^\mu(f) \leq n\Delta$. By Lemma 8, Adjust($\Delta$) transforms f to a $\Delta/2$-conservative solution by increasing the excess by at most $\frac{3}{2}m\Delta$. Hence the $\Delta/2$ phase starts with $Ex_\Delta^\mu(f) \leq (2n+3m)\Delta/2$, proving the claim. □

Lemma 9. *A Δ-phase consists of at most $2n+3m$ iterations.*

Proof. As for the linear case, let $\Psi(i) = \lfloor e_i^\mu/\Delta \rfloor - d_i$ if $e_i^\mu > (d_i+1)\Delta$ and $\Psi(i) = 0$ otherwise. Consider the potential function $\Psi = \sum_{i\in V}\Psi(i)$. By Claim 8, $\Psi \leq 2n+3m$ holds at the beginning. In the relabeling steps, Ψ may only decrease, and in every path augmentation, it decreases by exactly 1. □

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. For a Δ-conservative μ, $M_i\kappa_i = -e_i^\mu$ holds for every node i with $e_i < 0$, because of $\mu_i = 1/M_i$. Consequently, κ_f is the total relabeled deficiency of the negative nodes. The next theorem shows that if $\Delta < \varepsilon/(2n+4m)$, then we have an ε-approximate solution at the end of the Δ-phase.

Theorem 5. *At the end of phase Δ, the actual f is $(2n+4m)\Delta$-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 Δ and Δ', the total decrease of the excess discrepancy $\kappa_f = \sum_{i\in V} M_i\kappa_i = -\sum_{i\in V:\ e_i<0} e_i^\mu$ during the path augmentations is bounded by $(2n+3m)(\Delta/2+\Delta/4+\cdots+\Delta/2^k) < (2n+3m)\Delta$. Further, the subroutine Adjust can decrease the excess discrepancy by $m(\Delta/2+\Delta/4+\cdots+\Delta/2^k) < m\Delta$. Since in the Δ'-phase we have a 0-discrepancy solution, the total discrepancy at the end of the Δ-phase is at most $(2n+4m)\Delta$, proving the theorem.

Assume now the procedure runs forever. For each $i \in V$, κ_i is decreasing and thus converges to some limit κ_i^*. Let $\kappa^* = \sum_{i\in V} M_i\kappa_i^*$. As above, the total decrease of the excess discrepancy after phase Δ is bounded by $(2n+4m)\Delta$, hence $\kappa_f \leq \kappa^* + (2n+4m)\Delta$. The proof finishes by constructing an optimal pseudoflow f^* with discrepancy κ^*.

Let $f^{(t)}$ denote the pseudoflow 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 $\mu_i^* = \lim_{t\to\infty}\mu_i^{(t)}$. For every $ij \in E$, $f_{ij}^{(t)}$ is a bounded sequence $(0 \leq f_{ij}^{(t)} \leq u_{ij})$. Consequently, we can choose an infinite sequence $T' \subseteq \mathbb{N}$ so that restricted to $t \in T'$, all sequences $f_{ij}^{(t)}$ converge; let f_{ij}^* denote the limit. We shall prove that f^* is an optimal pseudoflow with optimal labeling μ_i^*, completing the proof.

Let $V_\infty = \{i\colon \mu_i^* = \infty\}$. We claim that $V \backslash V_\infty \neq \varnothing$. Indeed, if i is Δ-negative in a certain phase, then $\mu_i = 1/M_i$, and once i becomes Δ-positive or neutral, it would never again become Δ-negative. Consequently, the set of Δ-negative nodes is decreasing. Once it becomes empty, we arrive at a 0-discrepancy solution. If it never becomes empty, then we have a set N^* that remains the set of Δ-negative nodes after a finite number of steps and thus $\mu_i^* = 1/M_i$ for $i \in N^*$.

Let e_i^* denote the excess of f^*. 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 $t \in T'$, $(d_i+2n+3m)\Delta^{(t)}\mu_i^{(t)} < e_i^{(t)}$ and thus $Ex_{\Delta^{(t)}}^\mu(f) > (2n+3m)\Delta^{(t)}$, a contradiction.

INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

We have to prove $\Gamma_{ij}^{\mu^*+}(f_{ij}^{*\mu^*}) \le 1$ whenever $ij \in E_{f^*}$. If $\mu_j^* = \infty$, then $\Gamma_{ij}^{\mu^*+}(f_{ij}^{*\mu^*}) = 0$. If $\mu_j^* < \infty$, then the definition (9) gives

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

Then $\Delta^{(t)}\mu_j^{(t)} \to 0$ and hence the first fraction converges to $\Gamma_{ij}^+(f_{ij}^*) = 1/\{\Gamma_{ij}^{-1}\}^+(\Gamma_{ij}^+(f_{ij}^*))$, and the second to μ_i^*/μ_j^*, leading to the conclusion using Claim 5. $\square$

THEOREM 6. *The above algorithm finds an ε-approximate solution to the symmetric concave generalized flow problem in time $O(m(m\sigma + n\log n)\log(MUm/\varepsilon))$, where σ is the time needed for a value oracle call.*

PROOF. The initial value of Δ is $MU + 1$, and we terminate if $\Delta < \varepsilon/(2n + 4m)$. Hence the total number of scaling phases is $O(\log(MUm/\varepsilon))$. The number of iterations in a phase is $O(m)$ by Lemma 9. For the subroutine Tighten-Label, we first query the $\Gamma_{ij}(f_{ij})$ and $\Gamma_{ij}^{-1}(\Gamma_{ij}(f_{ij}) + \Delta\mu_j)$ values for every $ij \in E_f^{\mu}(\Delta)$. This enables computing the $\theta_{\Delta}^{\mu}(ij)$ values. The subroutine can be implemented as a slightly modified version of Dijkstra's algorithm time using Fibonacci heaps as in Fredman and Tarjan [11]. For every $ij \in S$, we have to query $\Gamma_{ij}^{-1}(\Gamma_{ij}(f_{ij}) + \Delta\mu_j)$ once more, when j enters the set S, or at the termination of the algorithm, in order to identify the final set of arcs in $E_f(\Delta)$. The total running time of Tighten-Label is hence $O(m\sigma + n\log n)$ this dominates the running time of an iteration. Adjust(Δ) can be performed by querying the values $\Gamma_{ij}(f_{ij})$ and $\Gamma_{ij}^{-1}(\Gamma_{ij}(f_{ij}) + (\Delta/2)\mu_j)$ values for every $ij \in E_f^{\mu}(\Delta/2)$ and modifying every f_{ij} value by $\pm(\Delta/2)\mu_i$ if necessary. Hence Adjust(Δ) runs in time $O(m\sigma)$. The overall running time bound then follows. $\square$

7. Sink versions of the problems. In this section, we show how the algorithms in §§5 and 6 can be applied to solve the sink versions of the corresponding problems. For linear generalized flows, let us set $M_t = 1$ and $M_i = B^n + 1$ for every $i \in V - t$. Let us set $b_t = \lceil \sum_{j:\, jt\in E} \gamma_{jt}u_{jt} - \sum_{j:\, tj\in E} l_{tj} \rceil + 1 \le d_t B^2 + 1$ and keep the same b_i for $i \ne t$. For an arbitrary pseudoflow, b_t is a strict upper bound on $\sum_{j:\ jt\in E} \gamma_{jt}f_{jt} - \sum_{j:\, tj\in E} f_{tj}$, hence $e_t < 0$ must hold.

Let us run the algorithm for the symmetric formulation with these M_i's and b_i's, returning an optimal pseudoflow f and optimal labels μ. 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 as f is feasible for the sink formulation and μ satisfies the dual optimality conditions (see §4.2). On the other hand, we prove the following.

LEMMA 10. *If there exists any node $i \in V - t$ with $e_i < 0$, then the sink version is infeasible.*

PROOF. Let S_0 be the set of nodes $i \in V - t$ with $e_i < 0$, and let S be the set of nodes $j \in V$ for which there exists a path in E_f from j to a node in S_0 ($S_0 \subseteq S \subseteq V$). We first show that $t \notin S$. Indeed, if P were a t–i path in E_f with $i \in S_0$, then $1 \ge \gamma^{\mu}(P) = \gamma(P)\mu_t/\mu_i$. Since both i and t are negative nodes, $\mu_i = 1/M_i = 1/(B^n + 1)$ and $\mu_t = 1$. Consequently, $\gamma(P) \le 1/(B^n + 1)$. This is a contradiction since $\gamma(P)$ is the product of at most $n - 1$ rational numbers, each with denominator at most B. Further, S may contain no nodes j with $e_j > 0$ by Lemma 1.

For a contradiction, assume g is a feasible solution to the sink version; choose g such that $\sum_{ij\in E} |f_{ij} - g_{ij}|$ is minimal. Let $e_i(g)$ denote the e_i value for g. By feasibility, $e_j(g) \ge 0 \ge e_j$ for every $j \in S$. By the definition of S, $g_{ij} \le u_{ij} = f_{ij}$ for every $ij \in E$ entering S and $g_{ij} \ge l_{ij} = f_{ij}$ for every $ij \in E$ leaving S. Since $S_0 \ne \varnothing$ and $e_i(g) > e_i$ for every $i \in S_0$, it is easy to see that there exists a directed cycle $C \subseteq E_f$ of arcs inside S with $f_{ij} < g_{ij}$ for every $ij \in C$. We have $\gamma_{ij}^{\mu} \le 1$ for all $ij \in E_f$ and therefore $\gamma(C) \le 1$. Consequently, we can decrease the g_{ij} values around C by a small amount without decreasing the $e_g(i)$ values for any node $i \in V$, and thereby we obtain another feasible solution. But this contradicts the extreme choice of g. $\square$

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 $O(m^2(m + n\log n)\log B)$.

Let us turn to concave generalized flows. An ε-approximate solution to the sink version means a pseudoflow f with $\sum_{i\in V-t} \max\{0, -e_i\} \le \varepsilon$ and e_t being at least the optimum value minus ε.

Let us set $b_t = U^* + 1$, and keep the same b_i for $i \ne t$. Now b_t is a strict upper bound on $\sum_{j:\, jt\in E} \Gamma_{jt}(f_{jt}) - \sum_{j:\, tj\in E} f_{tj}$ (recall the definition of U^* in §2.1). Thus $e_t < 0$ is guaranteed for every pseudoflow. Let us set $M_i = \lceil (2U^* + 1)/\varepsilon \rceil + 1$ if $i \in V - t$ and $M_t = 1$. Let us run the algorithm with these M_i and b_i values for the symmetric formulation to obtain an ε-optimal solution f.

If $\kappa_f > 2U^* + 1 + \varepsilon$, then we claim that no feasible solution exists for the sink version. Indeed, by the definition of U^*, if there is a feasible solution f', then there exists one with $e_t \ge -U^*$. If f' is such a feasible solution for the sink formulation, then its excess discrepancy for the symmetric formulation is at most $\kappa_{f'} \le b_t + U^* \le 2U^* + 1$, a contradiction as f was ε-optimal for the symmetric formulation.

INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

If $\kappa_f \le 2U^* + 1 + \varepsilon$, then we claim that f is ε-optimal for the sink version. Indeed,

$$\sum_{i\in V-t} \max\{0, -e_i\} = \frac{1}{\lceil (2U^*+1)/\varepsilon \rceil + 1} \sum_{i\in V-t} M_i \kappa_i \le \frac{\kappa_f}{\lceil (2U^*+1)/\varepsilon \rceil + 1} \le \varepsilon.$$

It is left to show that e_t cannot be further than ε from the optimum value in the sink formulation. Indeed, let f' be the optimal solution to the sink formulation with objective value e'_t. Note that f' is also a feasible solution to the symmetric formulation with $\kappa_{f'} = b_t - e'_t$. The claim follows by

$$b_t - e'_t + \varepsilon = \kappa_{f'} + \varepsilon \ge \kappa_f \ge \kappa_t = b_t - e_t,$$

and thus $e_t \ge e'_t - \varepsilon$. In the first inequality, we use that f is ε-optimal for the sink formulation. 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 that 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 nonsymmetric Arrow-Debreu Nash bargaining. 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) ε, the ε-approximate version is also infeasible. Similar reductions should hold for all other rational convex programs discussed in §3 as well, giving polynomial time algorithms for finding optimal solutions.

Theorem 7. *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 ε, T and a function $\tau(n, m, U^*)$.*

(P1) *Consider the algorithm for the sink version for an ε-approximation. Then either there exists no feasible solution, or $\mu_i \le T$ holds for any $i \in V$, even if running the algorithm for an arbitrary number of phases.*

(P2) *A subroutine is provided for finding an optimal solution $\tilde f$ in $\tau(n, m, U^*)$ time, if the following assumptions hold. Assume that for each $ij \in E$, we are given an interval $I_{ij} \subseteq [l_{ij}, u_{ij}]$ with $|I_{ij}| \le 2T\varepsilon$, 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 for finding the exact optimal solution or proving that the problem is infeasible in $O(m(m + n\log n)\log(U^* m/\varepsilon)) + \tau(n, m, U^*)$ time.

We remark that in (P2), $\tilde f = f^*$ is not required.

Proof. Let us formulate the symmetric version for ε-approximation as in §7. Assume we run the algorithm for this problem forever, as in the proof of Theorem 5. The μ_j's shall converge to some finite values $\mu^*_j \le T$ as otherwise infeasibility is implied by (P1). In any Δ-phase, the total change of f^μ_{ij} is bounded by $\varepsilon' = (2n+3m)\Delta$, and thus f_{ij} 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 5.

The algorithm terminates whenever $\Delta < \varepsilon/(2n+3m)$ or if $\mu_i > T$ for any $i \in V$—in the latter case we may conclude infeasibility. Assume now $\mu_i \le T$ for every $i \in V$ at termination. At this point, the intervals $I_{ij} = [f_{ij} - T\varepsilon, f_{ij} + T\varepsilon]$ satisfy the conditions in (P2), since $|f_{ij} - f^*_{ij}| \le T\varepsilon$. Running the ε-approximation algorithm and then the subroutine described in (P2) gives the running time bound. ☐

To ensure property (P2), a useful method is to enforce the existence of a unique optimal solution by perturbing the input data, as done by Orlin [31] 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 to nonsymmetric Arrow-Debreu Nash bargaining. Let us now apply Theorem 7 for the nonsymmetric ADNB problem. Let us assume all utilities U_{ij}, budgets m_i, and disagreement utilities c_i are nonnegative integers, with $U_{\max} = \max\{U_{ij}\colon i \in B, j \in G\}$, $R = \max\{m_i\colon i \in B\}$, and $C = \max\{c_i\colon i \in B\}$. Let $n = |G| + |B|$ and let m be the number of pairs ij with $U_{ij} > 0$. Let us assume that there exists at least one arc with positive utility incident to any buyer and to any good. The special case $c \equiv 0$ is identical to Fisher's market with linear utilities.

Consider a candidate solution with price p_j for each good $j \in G$. Let $x_{ij} \ge 0$ denote the amount of good j purchased by buyer i. It follows from the KKT-conditions (see also Vazirani [42]) that (p, x) is an optimal

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

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, $U_{ij}/p_j \le (\sum_{j\in G} U_{ij}x_{ij}-c_i)/m_i$, and equality holds whenever $x_{ij}>0$.

As shown in Vazirani [42] and discussed in the introduction, the market equilibria coincide with the optimal solutions to the convex program, a modification of (EG):

$$\begin{gathered}
\max \sum_{i\in B} m_i \log(z_i-c_i)\\
z_i \le \sum_{j\in G} U_{ij}x_{ij} \quad \forall\, i\in B\\
\sum_{i\in B} x_{ij}\le 1 \quad \forall\, j\in G\\
x\ge 0\\
z_i\ge c_i \quad \forall\, i\in B.
\end{gathered}$$

Unlike (EG), this problem might not have a feasible solution. 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 whenever the problem is feasible. Yet these same values can be obtained by different allocations. As in Orlin [31], we assume that there exists 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.

After the transformation described at the beginning of §6.1, we obtain the following concave generalized flow instance on $n+1$ nodes and $m+|B|$ arcs. The graph (V,E) is defined on the node set $V=B\cup G\cup\{t\}$. Let $ji\in E$ whenever $j\in G$, $i\in B$, $U_{ij}>0$, and set $\Gamma_{ji}(\alpha)=U_{ij}\alpha$ and $u_{ji}=2$. Add an arc $it\in E$ for every $i\in B$ with $\Gamma_{it}(\alpha)=m_i\log\alpha$ and $u_{it}=2\sum_{j\in G}U_{ji}$. Let $b_j=-1$ for $j\in G$, and $b_i=c_i$ for $i\in B$. The arc capacities are set in a way that the capacity constraints would never become tight. The complexity parameter U is bounded by $2|G|U_{\max}$. We shall prove the following.

Theorem 8. *Let $K=nRU_{\max}$. Setting $T=U^*=\max\{C, nK\log K\}$, $\varepsilon=1/(2K^nU^*)$ satisfies the requirements on U^* in §2.1 and* (P1) *and* (P2) *in Theorem 7. Our algorithm delivers an optimal solution in running time $O(m(m+n\log n)(n\log(nRU_{\max})+\log C))$ for nonsymmetric ADNB.*

The running time $\tau(n,m,U^*)$ will be negligible compared to the main algorithm and therefore it does not affect the complexity bound. Also note that our general algorithm assumes exact arithmetics and exact values returned by the oracles. Applying it directly would assume that we are able to compute the exact values of logarithmic functions. However, with some further technical work the algorithm can be transformed to a truly polynomial one, using only rational numbers of size polynomially bounded in the input size. In the rest of the section, we shall verify the choices of the parameters in the theorem.

Lemma 11 (See Vazirani [42, Thm 2], Orlin [31, Lemma 2.1]). *Assuming that the problem is feasible and there exists a unique optimal allocation x^*, all positive x^*_{ij} values are rational numbers with a common denominator $S\le K^n$.*

Proof. The optimal allocations x^* and prices p^* can be uniquely obtained given the set F of arcs ji with $x^*_{ij}>0$. If we introduce the variable $q_j=1/p_j$, then an optimal solution must satisfy the following system of linear equations:

$$\begin{gathered}
\sum_{j:\, ji\in E} x_{ij}=1 \quad \forall\, i\in G,\\
\sum_{k:\, ki\in F} U_{ik}x_{ik}-U_{ij}m_iq_j=c_i \quad \forall\, ji\in F.
\end{gathered}\tag{15}$$

We claim that this system has a unique solution $(x^*, 1/p^*)$. For every good j, $U_{ij}/p_j=(\sum_{k:\,ki\in F}U_{ik}x_{ik}-c_i)/m_i$ for every $ji\in F$; let us denote this common best bang-per-buck value by b_i. Let us set the price of an arbitrary good j_0 as $p_{j_0}=\alpha$ in an (undirected) connected component of F. Using that U_{ij}/p_j is the same on any two arcs in F incident to any $i\in B$, it follows that the value of α uniquely determines every p_j value in the same component, and p_j will be proportional to α. Consequently, all b_i values for $i\in B$ in the same component are proportional to $1/\alpha$. The optimality conditions imply that in an equilibrium, the money spent by buyer i is $r_i=m_i+c_i/b_i$, a linear function in α. In each component of F, the sum of prices should be equal to the

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

money spent by the buyers. This gives a linear equation on α hence uniquely determines all prices and bang-per-buck values in the component. The x_{ij} values in the component have to sum up to 1 for each good i and $\sum_{k:\,ki\in F} U_{ik}x_{ik} = b_i m_i + c_i$. As F is a forest, this system has a unique solution.

In the solution to (15), a common denominator is the determinant S of a largest nonsingular submatrix of the constraint matrix. The Hadamard bound gives $S \le (nRU_{\max})^n = K^n$, using that $|F| \le n-1$. □

The above proof also gives a simple linear time algorithm for finding the optimal solution, verifying (P2) if $2T\varepsilon < 1/K^n$, with $\tau(n, m, U^*)$ being negligible compared to the running time of the approximation algorithm.

Next we justify the choice of U^*. The inequality $U \le U^*$ clearly holds. For an arbitrary pseudoflow f, $e_t \le \sum_{i\in G} m_i \log v_{it} \le nR\log(nU) \le U^*$. It is left to show that if the problem is feasible, there exists a feasible solution with $e_t > -U^*$. Since the U_{ij} and c_i values are integers, whenever $\sum_{j:\,ji\in E} U_{ij}x_{ij} - c_i > 0$, it should be at least $1/K^n$. Consequently, if the sink version of the problem is feasible, the optimal objective value is at least $e_t \ge \sum_{i\in G} m_i \log(1/K^n) \ge -n^2R\log K \ge -U^*$.

The next claim verifies (P1) and thus completes the proof of Theorem 8.

Lemma 12. *Either the problem is not feasible, or $\mu_k \le U^*$ holds for any $k \in B\cup G$ in arbitrary Δ-phase.*

Proof. Recall from §7 that we solve the sink version by reducing it to the symmetric algorithm with $M_t = 1$ and $M_i = \lceil 2U^*/\varepsilon\rceil + 1$. Since μ_i is nondecreasing, these values converge to some limits $\mu_i^* \in \mathbb{R}\cup\{\infty\}$. We have $0 \le f_{ij} \le u_{ij}$ on all arcs $ij \in E$ in every phase, and therefore we can choose an infinite subset $T' \subseteq \mathbb{N}$ so that all f_{ij}'s converge if we restrict ourselves to iteration numbers in T'. As in Theorem 5, it can be easily verified that the limit f^* is an optimal solution to the symmetric version with conservative labeling μ^*.

As in §7, if $\kappa_f > 2U^* + 1$, then the sink version is not feasible, and otherwise f^* is also optimal to the sink version. In the latter case, $f_{it}^* > 0$ for arbitrary $i \in B$ must hold as otherwise $e_t = -\infty$ in the sink version gives infeasibility. Recall also that the symmetric version was defined such that $e_t < 0$ holds for every feasible solution and therefore $\mu_t^* = 1$.

Consider now an arbitrary $i \in B$. Both $it, ti \in E_f$ (it is easy to verify that $f_{it} = v_{it}$ is impossible), and therefore $(\mu_i^*/\mu_t^*)(m_i/f_{it}^*) = 1$ must hold by the conservativeness of μ^*. This means $\mu_i^* = f_{it}^*/m_i \le U \le U^*$.

Finally, let $j \in G$. Then for an arbitrary arc $ji \in E$, $ji \in E_f$ easily follows, and therefore conservativeness gives $(\mu_j^*/\mu_i^*)U_{ji} \le 1$, which implies $\mu_j^* \le U^*$. □

Finally, we remark that if we apply this algorithm to linear Fisher markets ($c \equiv 0$), the algorithm runs in a fundamentally different way as Devanur et al. [7] or Orlin [31]. Whereas 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. Our algorithm is not strongly polynomial. In fact, no such algorithm is known even for the linear case: it is a fundamental open question to find a strongly polynomial algorithm for linear generalized flows. If resolved, a natural question could be to devise a strongly polynomial algorithm for some class of convex generalized flow problems, analogously to the recent result (Végh [43]), desirably including the market and Nash bargaining applications.

Linear Fisher market is also captured by Végh [43]. A natural question is if there is any direct connection between our model and the convex minimum cost flow model studied in Végh [43]. Despite certain similarities, no reduction is known in any direction. Indeed, no such reduction is known even between the linear special cases, that is, generalized flows and minimum-cost circulations. In fact, the only known market setting captured by both is linear Fisher. Perfect price discrimination and ADNB are not known to be reducible to flows with convex objective. In contrast, spending constraint utilities (Vazirani [41]) are not known to be captured by our model, although they are captured by the other.

As discussed in §4.4, it seems difficult to extend any generalized flow algorithm having separate cycle cancelling and flow transportation subroutines. Although this includes the majority of combinatorial algorithms, there are some exceptions. Goldberg et al. [14] gave two different algorithms: besides Fat-Path, they also presented another 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 Hochbaum and Shantikumar [19] or Karzanov and McCormick [25] as a black box.

Another approach that avoids scaling is Wayne [44] for minimum-cost generalized flows and Restrepo and Williamson [34] 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. Although it does not seem easy, it might be possible to develop such an algorithm for concave generalized flows as well.

In defining an ε-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. Although the value oracle model as we use it, seems to need feasibility error, it might be possible to avoid it using a stronger oracle model as in Karzanov and McCormick [25]. One might 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. This work was done in the College of Computing, Georgia Institute of Technology, supported by the National Science Foundation [Grant CCF-0914732].

References

[1] Ahlfeld DP, Mulvey JM, Dembo RS, Zenios SA (1987) Nonlinear programming on generalized networks. *ACM Trans. Math. Software* 13(4):350–367.
[2] Ahuja RK, Magnanti TL, Orlin JB (1993) *Network Flows*: *Theory, Algorithms, and Applications* (Prentice-Hall, Upper Saddle River, NJ).
[3] Bertsekas DP, Polymenakos LC, Tseng P (1997) An ε-relaxation method for separable convex cost network flow problems. *SIAM J. Optim.* 7(3):853–870.
[4] Cohen E, Megiddo N (1994) New algorithms for generalized network flows. *Math. Programming* 64(1):325–336.
[5] Dantzig GB (1963) *Linear Programming and Extensions* (Princeton University Press, Princeton, NJ).
[6] Devanur NR, Jain K (2012) Online matching with concave returns. *Proc. 44th Sympos. Theory Comput.* (*STOC*), (ACM, New York), 137–144.
[7] Devanur NR, Papadimitriou CH, Saberi A, Vazirani VV (2008) Market equilibrium via a primal–dual algorithm for a convex program. *J. ACM* 55(5), Article 22.
[8] Edmonds J, Karp RM (1972) Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* 19(2):248–264.
[9] Eisenberg E, Gale D (1959) Consensus of subjective probabilities: The pari-mutuel method. *Ann. Math. Statist.* 30(1):165–168.
[10] Fleischer LK, Wayne KD (2002) Fast and simple approximation schemes for generalized flow. *Math. Programming* 91(2):215–238.
[11] Fredman ML, Tarjan RE (1987) Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM* 34(3):596–615.
[12] Goel G, Vazirani VV (2011) A perfect price discrimination market model with production, and a (rational) convex program for it. *Math. Oper. Res.* 36(4):762–782.
[13] Goldberg AV, Tarjan RE (1989) Finding minimum-cost circulations by canceling negative cycles. *J. ACM* 36(4):873–886.
[14] Goldberg AV, Plotkin SA, Tardos É (1991) Combinatorial algorithms for the generalized circulation problem. *Math. Oper. Res.* 16(2):351–381.
[15] Goldfarb D, Jin Z (1996) A faster combinatorial algorithm for the generalized circulation problem. *Math. Oper. Res.* 21(3):529–539.
[16] Goldfarb D, Lin Y (2002) Combinatorial interior point methods for generalized network flow problems. *Math. Programming* 93(2): 227–246.
[17] Goldfarb D, Jin Z, Lin Y (2002) A polynomial dual simplex algorithm for the generalized circulation problem. *Math. Programming* 91(2):271–288.
[18] Goldfarb D, Jin Z, Orlin JB (1997) Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem. *Math. Oper. Res.* 22(4):793–802.
[19] Hochbaum DS, Shanthikumar JG (1990) Convex separable optimization is not much harder than linear optimization. *J. ACM* 37(4):843–862.
[20] Jain K, Vazirani VV (2010) Eisenberg-Gale markets: Algorithms and game-theoretic properties. *Games Econom. Behav.* 70(1):84–106.
[21] Jewell WS (1962) Optimal flow through networks with gains. *Oper. Res.* 10(4):476–499.
[22] Kalai E (1977) Nonsymmetric Nash solutions and replications of 2-person bargaining. *Internat. J. Game Theory* 6(3):129–133.
[23] Kantorovich LV (1939) Mathematical methods of organizing and planning production. *Publication House of the Leningrad State University* 68, 6(4):366–422 [English translation in *Management Science]*.
[24] Kapoor S, Vaidya PM (1996) Speeding up Karmarkar's algorithm for multicommodity flows. *Math. Programming* 73(1):111–127.
[25] Karzanov AV, McCormick ST (1997) Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM J. Comput.* 26(4):1245–1275.
[26] Minoux M (1986) Solving integer minimum cost flows with separable convex cost objective polynomially. Gallo G, Sandi C, eds. *Netflow at Pisa*, Math. Programming Stud., Vol. 26 (Springer, Berlin, Heidelberg), 237–239.
[27] Nisan N, Roughgarden T, Tardos E, Vazirani VV (2007) *Algorithmic Game Theory* (Cambridge University Press, New York).
[28] Onaga K (1966) Dynamic programming of optimum flows in lossy communication nets. *IEEE Trans. Circuit Theory* 13:308–327.
[29] Onaga K (1967) Optimum flows in general communication networks. *J. Franklin Institute* 283(4):308–327.
[30] Orlin JB (1993) A faster strongly polynomial minimum cost flow algorithm. *Oper. Res.* 41(2):338–350.
[31] Orlin JB (2010) Improved algorithms for computing Fisher's market clearing prices. *Proc. 42nd ACM Sympos. Theory Comput. (STOC)* (ACM, New York), 291–300.
[32] Radzik T (1993) Approximate generalized circulation. Technical Report 93-2, Cornell Computational Optimization Project, Cornell University, Ithaca, NY.
[33] Radzik T (2004) Improving time bounds on maximum generalised flow computations by contracting the network. *Theoret. Comput. Sci.* 312(1):75–97.
[34] Restrepo M, Williamson DP (2009) A simple GAP-canceling algorithm for the generalized maximum flow problem. *Math. Programming* 118(1):47–74.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

INFORMS holds **copyright** to this article and distributed this copy as a courtesy to the author(s).
Additional information, including rights and permission policies, is available at https://www.informs.org/.

[35] Shigeno M (2004) A survey of combinatorial maximum flow algorithms on a network with gains. *J. Oper. Res. Soc. Japan* 47:244–264.
[36] Shigeno M (2006) Maximum network flows with concave gains. *Math. Programming* 107(3):439–459.
[37] Shmyrev VI (2009) An algorithm for finding equilibrium in the linear exchange model with fixed budgets. *J. Appl. Indust. Math.* 3(4):505–518.
[38] Tardos É, Wayne KD (1998) Simple generalized maximum flow algorithms. Bixby RE, Boyd EA, Rios-Mercado RZ, eds. *Proc. 6th Internat. IPCO (Integer Programming and Combinatorial Optimization) Conf.*, Lecture Notes in Computer Science, Vol. 1412 (Springer, Berlin, Heidelberg), 310–324.
[39] Truemper K (1977) On max flows with gains and pure min-cost flows. *SIAM J. Appl. Math.* 32(2):450–456.
[40] Truemper K (1978) Optimal flows in nonlinear gain networks. *Networks* 8(1):17–36.
[41] Vazirani VV (2010) Spending constraint utilities with applications to the Adwords market. *Math. Oper. Res.* 35(2):458–478.
[42] Vazirani VV (2012) The notion of a rational convex program, and an algorithm for the Arrow-Debreu Nash bargaining game. *J. ACM* 59(2), Article 7.
[43] Végh LA (2012) Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. *Proc. 44th Sympos. Theory Comput. (STOC)*, (ACM, New York), 27–40.
[44] Wayne KD (2002) A polynomial combinatorial algorithm for generalized minimum cost flow. *Math. Oper. Res.* 27(3):445–459.