László A. Végh

A strongly polynomial algorithm for generalized flow maximization

A strongly polynomial algorithm for generalized flow maximization

László A. Végh

Department of Mathematics
London School of Economics and Political Science
L.Vegh@lse.ac.uk

Abstract

A strongly polynomial algorithm is given for the generalized flow maximization problem. It uses a new variant of the scaling technique, called continuous scaling. The main measure of progress is that within a strongly polynomial number of steps, an arc can be identified that must be tight in every dual optimal solution, and thus can be contracted. As a consequence of the result, we also obtain a strongly polynomial algorithm for the linear feasibility problem with at most two nonzero entries per column in the constraint matrix.

1 Introduction

The generalized flow model is a classical extension of network flows. Besides the capacity constraints, for every arc e there is a gain factor $\gamma_e > 0$, such that flow amount gets multiplied by γ_e while traversing arc e. We study the flow maximization problem, where the objective is to send the maximum amount of flow to a sink node t. The model was already formulated by Kantorovich [19], as one of the first examples of Linear Programming; it has several applications in Operations Research [2, Chapter 15]. Gain factors can be used to model physical changes such as leakage or theft. Other common applications use the nodes to represent different types of entities, e.g. different currencies, and the gain factors correspond to the exchange rates.

The existence of a strongly polynomial algorithm for Linear Programming is a major open question in the theory of computation. This refers to an algorithm with the number of arithmetic operations polynomially bounded in the number of variables and constraints, and the size of the numbers during the computations polynomially bounded in the input size. A landmark result by Tardos [30] is an algorithm with the running time dependent only on the size of numbers in the constraint matrix, but independent from the right-hand side and the objective vector. This gives strongly polynomial algorithms for several combinatorial problems such as minimum cost flows (see also Tardos [29]) and multicommodity flows.

Instead of bounding the sizes of numbers, one might impose structural restrictions on the constraint matrix. A natural question arises whether there exists a strongly polynomial algorithm for linear programs (LPs) with at most two nonzero entries per column (that can be arbitrary numbers). This question is still open; as shown by Hochbaum [17], all such LPs can be efficiently transformed to equivalent instances of the minimum cost generalized flow problem. (Note also that every LP can be efficiently transformed to an equivalent one with at most three nonzero entries per column.) In 1983, Megiddo [21] gave a strongly polynomial algorithm for solving the dual feasibility problem for such LPs; he introduced the concept of strongly polynomial algorithms in the same paper. A corollary of our result is the first strongly polynomial algorithm for the primal feasibility problem.

Generalized flow maximization is probably the simplest natural class of LPs where no strongly polynomial algorithm was known. The existence of such an algorithm has been a well-studied and longstanding open problem (see e.g. [9, 3, 35, 26, 28]). A strongly polynomial algorithm for a restricted class was given by Adler and Cosares [1].

In this paper, we exhibit a strongly polynomial algorithm for generalized flow maximization. Let n denote the number of nodes and m the number of arcs in the network, and let B denote the largest integer used in the description of the input (see Section 2 for the precise problem definition). A strongly polynomial algorithm for the problem entails the following (see [16]): *(i)* it uses only elementary arithmetic operations (addition, subtraction, multiplication, division), and comparisons; *(ii)* the number of these operations is bounded by a polynomial of n and m; *(iii)* all numbers occurring in the computations are rational numbers of encoding size polynomially bounded in n, m and $\log B$ – or equivalently, it is a polynomial space algorithm. Here, the encoding size of a positive rational number p/q is defined as $\lceil \log_2(p+1) \rceil + \lceil \log_2(q+1) \rceil$. By the running time of a strongly polynomial algorithm we mean the total number of elementary arithmetic operations and comparisons.

Combinatorial approaches have been applied to generalized flows already in the sixties by Dantzig [4] and Jewell [18]. However, the first polynomial-time combinatorial algorithm was only given in 1991 by Goldberg, Plotkin and Tardos [9]. This was followed by a multitude of further combinatorial algorithms e.g. [3, 11, 13, 31, 6, 12, 14, 35, 26, 27, 34]; a central motivation of this line of research was to develop a strongly polynomial algorithm. The algorithms of Cohen and Megiddo [3], Wayne [35], and Restrepo and Williamson [27] present fully polynomial time approximation schemes, that is, for every $\varepsilon > 0$, they can find a solution within ε from the optimum value in running time polynomial in n, m and $\log(1/\varepsilon)$. This can be transformed to an optimal solution for a sufficiently small ε; however, this value does depend on B and hence the overall running time will also depend on $\log B$. The current most efficient weakly polynomial algorithms are the interior point approach of Kapoor an Vaidya [20] with running time $O(m^{1.5}n^2 \log B)$, and the combinatorial algorithm by Radzik [26] with running time $\tilde{O}(m^2 n \log B)$.[1] For a survey on combinatorial generalized flow algorithms, see Shigeno [28].

The generalized flow maximization problem exhibits deep structural similarities to the minimum cost circulation problem, as first pointed out by Truemper [32]. Most combinatorial algorithms for generalized flows, including both algorithms by Goldberg et al. [9], exploit this analogy and adapt existing efficient techniques from minimum cost circulations. For the latter problem, several strongly polynomial algorithms are known, the first one given by Tardos [29]; others relevant to our discussion are those by Goldberg and Tarjan [10], and by Orlin [23]; see also [2, Chapters 9-11]. Whereas these algorithms serve as starting points for most generalized flow algorithms, the applicability of the techniques is by no means obvious, and different methods have to be combined. As a consequence, the strongly polynomial analysis cannot be carried over when adapting minimum cost circulation approaches to generalized flows, although weakly polynomial bounds can be shown. To achieve a strongly polynomial guarantee, further new algorithmic ideas are required that are specific to the structure of generalized flows. The new ingredients of our algorithm are highlighted in Section 2.4.

Let us now outline the scaling method for minimum cost circulations, a motivation of our generalized flow algorithm. The first (weakly) polynomial time algorithm for minimum cost circulations was given by Edmonds and Karp [5], introducing the simple yet powerful idea of scaling (see also [2, Chapter 9.7]). The algorithm consists of Δ-phases, with the value of $\Delta > 0$ decreasing by a factor of at least two between every two phases, yielding an optimal solution for sufficiently small Δ. In

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

the Δ-phase, the flow is transported in units of Δ from nodes with excess to nodes with deficiency using shortest paths in the graph of arcs with residual capacity at least Δ. Orlin [23], (see also [2, Chapters 10.6-7]) devised a strongly polynomial version of this algorithm. The key notion is that of "abundant arcs". In the Δ-phase of the scaling algorithm [5], the arc e is called *abundant* if it carries $> 4n\Delta$ units of flow. For such an arc e, it can be shown that $x^*_e > 0$ must hold for some optimal solution x^*. By primal-dual slackness, the corresponding constraint must be tight in every dual optimal solution. Based on this observation, Orlin [23] showed that such an arc can be contracted; the scaling algorithm is then restarted on the smaller graph. This leads to a dual optimal solution in strongly polynomial time; that provided, a primal optimal solution can be found via a single maximum flow computation. Orlin [23] also presents a more efficient but also more sophisticated implementation of this idea.

Let us now turn to generalized flows. The analogue of the scaling method was an important component of the Fat-Path algorithm of [9]; the algorithm of Goldfarb, Jin and Orlin [13] and the one in [34] also use this technique. The notion of "abundant arcs" can be easily extended to these frameworks: if an arc e carries a "large" amount of flow as compared to Δ, then it must be tight in every dual optimal solution, and hence can be contracted. This idea was already used by Radzik [26], to boost the running time of [13]. Nevertheless, it is not known whether an "abundant arc" would always appear in any of the above algorithms within a strongly polynomial number of steps.

Our contribution is a new type of scaling algorithm that suits better the dual structure of the generalized flow problem, and thereby the quick appearance of an "abundant arc" will be guaranteed. Whereas in all previous methods, the scaling factor Δ remains constant for a linear number of path augmentations, our *continuous scaling method* keeps it decreasing in every elementary iteration of the algorithm, even in those that lead to finding the next augmenting path.

The rest of the paper is structured as follows. Section 2 first defines the problem setting, introduces relabelings, gives the characterization of optimality, and defines the notion of Δ-feasibility. Section 2.4 then gives a more detailed account of the main algorithmic ideas.

The algorithm is presented in three different versions. First, Section 2.5 describes a relatively simple scaling algorithm called Continuous Scaling, with a weakly polynomial running time guarantee proved in Section 3. Our strongly polynomial algorithm Enhanced Continuous Scaling in Section 4 builds on this, by including one additional subroutine, and a framework for contracting arcs. The running time analysis is given in Section 5. This achieves a strongly polynomial bound on the number of steps. A strongly polynomial algorith must also satisfy requirement *(iii)* on bounded number sizes. This requires further modifications of the algorithm in Section 6 by introducing certain rounding steps.

Section 7 shows reductions between different formulations; in particular, the corollary on LP feasibility problems with at most two nonzeros per column is shown here. Section 7 is independent from the preceding sections and can be read directly after Section 2. Section 8 concludes with some additional remarks and open questions.

2 Preliminaries

We start by introducing the most general formulation our approach is applicable to. Consider the linear feasibility problem

$$\begin{aligned} Ax &= b \\ 0 \le x &\le u, \end{aligned} \tag{LP2}$$

such that every column of A contains at most two nonzero entries. By making use of a reduction by Hochbaum [17], in Section 7.2 we show that every problem of this form can be reduced to the generalized flow maximization problem as defined next.

Let $G = (V, E)$ be a directed graph with a designated sink node $t \in V$. Let $n = |V|$, $m = |E|$, and for each node $i \in V$, let d_i denote total number of arcs incident to i (both entering and leaving). We will always assume $n \le m$. We do not allow parallel arcs and hence we may use ij to denote the arc from i to j. This is for notational convenience only, and all results straightforwardly extend to the setting with parallel arcs. All paths and cycles in the paper will refer to directed paths and directed cycles.

The following is the *standard formulation* of the problem. Let us be given arc capacities $u : E \to \mathbb{Q}_{>0} \cup \{\infty\}$ and gain factors $\gamma : E \to \mathbb{Q}_{>0}$.

$$\begin{aligned} \max \sum_{j:jt\in E} \gamma_{jt} f_{jt} - \sum_{j:tj\in E} f_{tj} & \\ \sum_{j:ji\in E} \gamma_{ji} f_{ji} - \sum_{j:ij\in E} f_{ij} \ge 0 \quad & \forall i \in V - t \\ 0 \le f \le u & \end{aligned} \tag{P_u}$$

It is common in the literature to define the problem with equalities in the node constraints. The two forms are essentially equivalent, see e.g. [28]; moreover, the form with equality is often solved via a reduction to (P_u). In this paper, we prefer to use yet another equivalent formulation, where the arcs have no upper capacities, but there are node demands instead. A problem given in the standard formulation can be easily transformed to an equivalent instance in this form; the transformation is described in Section 7.1. Given a node demand vector $b : V \to \mathbb{Q}$ and gain factors $\gamma : E \to \mathbb{Q}_{>0}$, the *uncapacitated formulation* is defined as

$$\begin{aligned} \max \sum_{j:jt\in E} \gamma_{jt} f_{jt} - \sum_{j:tj\in E} f_{tj} & \\ \sum_{j:ji\in E} \gamma_{ji} f_{ji} - \sum_{j:ij\in E} f_{ij} \ge b_i \quad & \forall i \in V - t \\ 0 \le f & \end{aligned} \tag{P}$$

Note the value of b_t is irrelevant as it is not present in the formulation; we may e.g. assume $b_t = 0$. For a vector $f \in \mathbb{R}^{|E|}_{\ge 0}$, let us define the *excess* of a node $i \in V$ by

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

The node constraints in (P) can be written as $e_i(f) \ge 0$, and the objective is equivalent to maximizing $e_t(f)$. When f is clear from the context, we will denote the excess simply by $e_i := e_i(f)$. By a *generalized flow* we mean a feasible solution to (P), that is, a nonnegative vector $f \in \mathbb{R}^{|E|}_{\ge 0}$ with $e_i(f) \ge 0$ for all $i \in V - t$. Let us define the *surplus* of f as

$$Ex(f) := \sum_{i\in V-t} e_i(f).$$

It will be convenient to make the following assumptions.

There is an arc $it \in E$ for every $i \in V - t$; $(\star)$

The problem (P) is feasible, and an initial feasible solution $\bar{f}$ is provided. $(\star\star)$

The objective value in (P) is bounded. $(\star\star\star)$

These assumptions are without loss of generality; it is shown in Section 7.1 that any problem in the standard form can be transformed to an equivalent one in the uncapacitated form that also satisfies assumptions ($\star$) and ($\star\star$). Condition ($\star$) can be easily achieved by adding new arcs to the sink with gain factors small enough not to influence the solution. To obtain ($\star\star$), observe that $f \equiv 0$ is feasible to (P_u); $\bar{f}$ in ($\star\star$) will be the image of 0 under the transformation. To justify ($\star\star\star$), in the same Section 7.1 we show how unboundedness can be detected. Furthermore, in Section 7.2 we show how an arbitrary instance of (LP2) can be reduced to solving two instances of (P) satisfying these assumptions.

Let us introduce some further notation. For an arc set $H \subseteq E$, let $\overleftarrow{H}$ denote the set of reverse arcs, that is, $\overleftarrow{H} := \{ji : ij \in H\}$; let $\overleftrightarrow{H} := H \cup \overleftarrow{H}$. We define the gain factor of a reverse arc $ji \in \overleftarrow{H}$ by $\gamma_{ji} := 1/\gamma_{ij}$. For an arc set $F \subseteq E$ and node sets $S, T \subseteq V$, let $F[S,T] := \{ij \in F : i \in S, j \in T\}$. We also use $F[S] := F[S,S]$ to denote the set of arcs in F spanned by S. For a node $i \in V$, let $\delta^{in}(i)$ and $\delta^{out}(i)$ denote the set of arcs entering and leaving i, respectively. We will use the vector norms $||x||_1 = \sum_i |x_i|$ and $||x||_\infty = \max_i |x_i|$. For integers $a \leq b$, let $[a,b] := \{a, a+1, \dots, b\}$.

A vector $f : \overleftrightarrow{E} \to \mathbb{R}_{\geq 0}$ is called a *path flow*, if its support is a path $P = w_1 w_2 \dots w_t \subseteq \overleftrightarrow{E}$, and $\gamma_{w_\ell} f_{w_\ell} = f_{w_{\ell+1}}$ for every $1 \leq \ell \leq t-1$. In other words, the incoming flow equals the outgoing flow in every internal node of the path. We say that a path flow f sends α units of flow from p to q, if the support of f is a $p-q$ path, and the flow value arriving at q equals α. Note however, that the amount of flow leaving p is typically different from α.

2.1 Encoding size

In the weakly polynomial algorithm, the running time will be dependent on the encoding size of the input, that consists of rational numbers. In a strongly polynomial algorithm, all numbers appearing during the computations must be rational of encoding size polynomially bounded in the input size. (We remark that the notion of strongly polynomial algorithms is also applicable to problems with arbitrary real numbers in the input; this model assumes that every basic arithmetic operation can be carried out in $O(1)$ time.)

Standard formulation. We are given an integer B such that all capacities u and gain factors γ are rational numbers, given as quotients of two integers $\leq B$.

Uncapacitated formulation. We give more complicated conditions on the encoding size of the different quantities. This is in order to maintain good bounds on the encoding size when transforming an instance from the standard to the uncapacitated formulation in Section 7.1.

Assume the instance satisfies conditions ($\star$), ($\star\star$) and ($\star\star\star$). We use the integer $\bar{B}$ to bound the encoding size of the input as follows.

- The arcs can be classified into two types, *regular* and *auxiliary*, with t being the endpoint of every auxiliary arc. For a regular arc ij, the gain factor γ_{ij} is given as a rational number, such that $\bar{B}$ is an integer multiple of the product of the numerators and denominators of all γ_{ij} values for regular arcs. For every auxiliary arc it, $\gamma_{it} = 1/\bar{B}$.
- For every $i \in V - t$, $|b_i| \leq \bar{B}$, and is an integer multiple of $1/\bar{B}$.
- For the initial solution $\bar{f}$, and for every $ij \in E$, $\bar{f}_{ij} \leq \bar{B}$ and $\bar{f}_{ij}$ is an integer multiple of $1/\bar{B}$.

The reduction in Section 7.1 will transform an instance in the standard formulation with n nodes and m arcs and parameter B to an uncapacitated instance with $\leq m+n$ nodes, $\leq 2m$ arcs and $\bar{B} \leq 2B^{4m}$.

Our main result is the following.

Theorem 2.1. *There exists an $O(n^3m^2)$ time strongly polynomial algorithm for the uncapacitated formulation (P) with assumptions $(\star)$, $(\star\star)$ and $(\star\star\star)$.*

Using the transformation in Section 7.1, this gives an $O(m^5)$ time strongly polynomial algorithm for the standard formulation (P_u). Finally, using the reduction in Section 7.2, we get an $O(m^5)$ algorithm for the linear feasibility problem (LP2) with n constraints and m variables.

2.2 Labelings and optimality conditions

Dual solutions to (P) play a crucial role in the entire generalized flow literature. Let $y : V \to \mathbb{R}_{\geq 0}$ be a solution to the dual of (P). Following Glover and Klingman [8], the literature standard is not to consider the y values but their inverses instead. With $\mu_i := 1/y_i$, we can write the dual of (P) in the following form.

$$\begin{aligned} &\max \sum_{i \in V} \frac{b_i}{\mu_i} \\ &\gamma_{ij}\mu_i \leq \mu_j \quad \forall ij \in E \\ &\quad \mu_i > 0 \quad \forall i \in V - t \\ &\quad \mu_t = 1 \end{aligned} \tag{D}$$

A feasible solution μ to this program will be called a *relabeling* or *labeling*. An *optimal labeling* is an optimal solution to (D). Whereas there could be values $\mu_i = \infty$ corresponding to $y_i = 0$, assumption $(\star)$ guarantees that all μ_i values must be finite. A useful and well-known property is the following.

Proposition 2.2. *Given an optimal solution to (D), an optimal solution to (P) can be obtained in strongly polynomial time, and conversely, given an optimal solution to (P), an optimal solution to (D) can be obtained in strongly polynomial time.*

In fact, our strongly polynomial algorithm proceeds via finding an optimal solution to (D), and computing the primal optimal solution via a single maximum flow computation. The first part of the above proposition is proved in Theorem 2.6(i), whereas the second part (which is not needed for our algorithm) can be shown using an argument similar to the proof of Lemma 3.1.

Relabelings will be used in all parts of the algorithm and proofs. For a generalized flow f and a labeling μ, we define the relabeled flow f^μ by

$$f^\mu_{ij} := \frac{f_{ij}}{\mu_i}$$

for all $ij \in E$. This can be interpreted as changing the base unit of measure at the nodes (i.e. in the example of the currency exchange network, it corresponds to changing the unit from pounds to pennies). To get a problem setting equivalent to the original one, we have to relabel all other quantities accordingly. That is, we define relabeled gains, demands, excesses and surplus by

$$\gamma^\mu_{ij} := \gamma_{ij}\frac{\mu_i}{\mu_j}, \quad b^\mu_i := \frac{b_i}{\mu_i}, \quad e^\mu_i := \frac{e_i}{\mu_i}, \text{ and } \quad Ex^\mu(f) := \sum_{i \in V-t} e^\mu_i,$$

respectively. Another standard notion is the *residual network* $G_f = (V, E_f)$ of a generalized flow f, defined as

$$E_f := E \cup \{ij : ji \in E, f_{ji} > 0\}.$$

Arcs in E are called *forward arcs*, while arcs in the second set are *reverse arcs*. Recall that for a reverse arc ji we defined $\gamma_{ji} = 1/\gamma_{ij}$. Also, we define $f_{ji} := -\gamma_{ij} f_{ij}$ for every reverse arc $ji \in E_f$.

By increasing (decreasing) f_{ji} by α on a reverse arc $ji \in E_f$, we mean decreasing (increasing) f_{ij} by α/γ_{ij}. The input graph $G = (V, E)$ is allowed to have pairs of oppositely directed arcs ij and ji, making our notation slightly ambiguous: for an arc ij, we will denote its reverse arc by ji, which might be an arc parallel to the original arc from j to i in the input. However, this should not be a source of confusion: whenever the arc ji is mentioned in the context of ij, it will always refer to the reverse arc.

The crucial notion of conservative labelings is motivated by primal-dual slackness. Let f be a generalized flow (that is, a feasible solution to (P)), and let $\mu : V \to \mathbb{R}_{>0}$. We say that μ is a *conservative labeling* for f, if μ is a feasible solution to (D) with the further requirement that $\gamma^\mu_{ij} = 1$ whenever $f_{ij} > 0$ for $ij \in E$. The following characterization of optimality is a straightforward consequence of primal-dual slackness in Linear Programming. We state the optimality conditions both for the uncapacitated formulation (P), and for the standard formulation (P_u). In the latter part we do not assume $(\star)$, and therefore $\mu_i = \infty$ is also allowed.

Theorem 2.3. *(i) Assume $(\star)$ holds. A generalized flow f is an optimal solution to (P) if and only if there exists a finite conservative labeling μ, and $e_i = 0$ for all $i \in V - t$.*

(ii) A feasible solution f to the standard form (P_u) is optimal if and only if there exists a function $\mu : V \to \mathbb{R}_{>0} \cup \{\infty\}$ such that $\mu_t = 1$, and $\gamma_{ij}\mu_i \le \mu_j$ if $f_{ij} = 0$, $\gamma_{ij}\mu_i = \mu_j$ if $0 < f_{ij} < u_{ij}$, and $\gamma_{ij}\mu_i \ge \mu_j$ if $f_{ij} = u_{ij}$; further, $e_i = 0$ whenever $\mu_i < \infty$.

Given a labeling μ, we say that an arc $ij \in E_f$ is *tight* if $\gamma^\mu_{ij} = 1$. A directed path in E_f is called *tight* if it consists of tight arcs.

2.3 Δ-feasible labels

Let us now introduce a relaxation of conservativity crucial in the algorithm. This is new notion, although similar concepts have been used in previous scaling algorithms [11, 34]. Section 2.4 explains the background and motivation of this notion. Given a labeling μ, let us call arcs in E with $\gamma^\mu_{ij} < 1$ *non-tight*, and denote their set by

$$F^\mu := \{ij \in E : \gamma^\mu_{ij} < 1\}.$$

For every $i \in V$, let

$$R_i := \sum_{j:ji \in F^\mu} \gamma_{ji} f_{ji}$$

denote the total flow incoming on non-tight arcs; let $R^\mu_i := \frac{R_i}{\mu_i} = \sum_{j:ji\in F^\mu} \gamma^\mu_{ji} f^\mu_{ji}$. For some $\Delta \ge 0$, let us define the Δ-*fat graph* as

$$E^\mu_f(\Delta) = E \cup \{ij : ji \in E, f^\mu_{ji} > \Delta\}.$$

We say that μ is a Δ-*conservative labeling* for f, or that (f, μ) is a Δ-*feasible pair*, if

- $\gamma^\mu_{ij} \le 1$ holds for all $ij \in E^\mu_f(\Delta)$, and
- $\mu_t = 1$, and $\mu_i > 0$, $e_i \ge R_i$ for every $i \in V - t$.

Note that in particular, μ must be a feasible solution to (D). The first condition is equivalent to requiring $f^\mu_{ij} \le \Delta$ for every non-tight arc. Note that 0-conservativeness is identical to conservativeness: $E^\mu_f(0) = E^\mu_f$, and therefore every arc carrying positive flow must be tight; the second condition simply gives $e_i \ge 0$ whenever $\mu_i > 0$. The next lemma can be seen as the converse of this observation.

Lemma 2.4. *Let (f, μ) be a Δ-feasible pair for some $\Delta > 0$. Let us define the generalized flow $\tilde{f}$ with $\tilde{f}_{ij} = 0$ if $ij \in F^\mu$ and $\tilde{f}_{ij} = f_{ij}$ otherwise. Then $\tilde{f}$ is a feasible generalized flow, μ is a conservative labeling for $\tilde{f}$, and $Ex^\mu(\tilde{f}) \le Ex^\mu(f) + |F^\mu|\Delta$.*

Proof. For feasibility, we need to verify $e_i(\tilde{f}) \ge 0$ for all $i \in V - t$. This follows since

$$e_i(\tilde{f}) \ge e_i(f) - \sum_{j:ji \in F^\mu} \gamma_{ji} f_{ji} = e_i(f) - R_i \ge 0.$$

It is straightforward by the construction that $\gamma^\mu_{ij} \le 1$ for every $ij \in E$ with equality whenever $\tilde{f}_{ij} > 0$. This shows that μ is a conservative labeling. For the last part, observe that decreasing the flow value to 0 on a non-tight arc ij may create $f^\mu_{ij} \le \Delta$ units of relabeled excess at i. □

Claim 2.5. *In a Δ-conservative labeling, $R^\mu_i < d_i\Delta$ holds for every $i \in V$.*

Proof. If μ is a Δ-conservative labeling, then $f^\mu_{ji} \le \Delta$ holds for every non-tight arc ji; also note that the relabeled flow arriving from j on a non-tight arc is $\gamma^\mu_{ji} f^\mu_{ji} < f^\mu_{ji} \le \Delta$, and hence $R^\mu_i < d_i\Delta$. □

2.4 Overview of the algorithms

We now informally describe some fundamental ideas of our algorithms Continuous Scaling and Enhanced Continuous Scaling, and explain their relations to previous generalized flow algorithms. The precise algorithms and arguments will be given in the later sections.

Basic features of the algorithms

Given a generalized flow f, a cycle C in the residual graph E_f is called *flow-generating*, if $\gamma(C) = \prod_{e \in C} \gamma_e > 1$. If there exists a flow-generating cycle, then some positive amount of flow can be sent around it to create positive excess in an arbitrary node i incident to C.

The notion of conservative labellings is closely related to flow generating cycles. Notice that for an arbitrary labeling μ, $\gamma(C) = \gamma^\mu(C)$. Therefore, if μ is a finite conservative labeling, then E_f cannot contain any flow-generating cycles. It is also easy to verify the converse: if there are no flow-generating cycles, then there exists a conservative labeling (see also Lemma 3.1).

The Maximum-mean-gain cycle-canceling procedure, introduced by Goldberg et al. [9], can be used to eliminate all flow-generating cycles efficiently. The subroutine proceeds by choosing a cycle $C \subseteq E_f$ maximizing $\gamma(C)^{1/|C|}$, and from an arbitrary node i incident to C, sending the maximum possible amount of flow around C admitted by the capacity constraints, thereby increasing the excess e_i. It terminates once there are no more flow-generating cycles left in E_f. This is a natural analogue of the minimum mean cycle cancellation algorithm of Goldberg and Tarjan [10] for minimum cost circulations. Radzik [25] (see also [28]) gave a strongly polynomial running time bound $O(m^2 n \log^2 n)$ for the Maximum-mean-gain cycle-canceling algorithm.

Our algorithm also starts with performing this algorithm, with the input being the initial solution $\bar{f}$ provided by ($\star\star$). Hence one can obtain a feasible solution f along with a conservative labeling μ in strongly polynomial time.

Such an f can be transformed to an optimal solution using Onaga's algorithm [22]: while there exists a node $i \in V - t$ with $e_i > 0$, find a *highest gain augmenting path* from i to t, that is, a path P in the residual graph E_f with the product of the gains maximum. Send the maximum amount of flow on this augmenting path enabled by the capacity constraints. A conservative labeling can be used to identify such paths: we can transform a conservative labeling to a *canonical labeling* (see [9]), where every node i is connected to the sink via a tight path. Such a canonical labeling can be found via a Dijkstra-type algorithm, increasing the labels of certain nodes. The correctness of Onaga's algorithm follows by the observation that sending flow on a tight path maintains the conservativeness of the labeling, hence no new flow-generating cycles may appear.

Unfortunately, Onaga's algorithm may run in exponentially many steps, and might not even terminate if the input is irrational. The Fat-Path algorithm [9] introduces a scaling technique to overcome this difficulty. The algorithm maintains a scaling factor Δ that decreases geometrically. In the Δ-phase, flow is sent on a highest gain "Δ-fat" augmenting path, that is, a highest gain path among those that have sufficient capacity to send Δ units of flow to the sink. In our notation, these are paths in $E_f^\mu(\Delta)$. However, path augmentations might create new flow-generating cycles, which have to be repeatedly cancelled by calling the cycle-canceling subroutine at the beginning of every phase.

Our notion of Δ-feasible pairs in Section 2.3 is motivated by the idea of Δ-fat paths: note that every arc in the Δ-fat graph $E_f^\mu(\Delta)$ has sufficient capacity to send Δ units of relabeled flow. A main step in our algorithm will be sending Δ units of relabeled flow on a tight path in $E_f(\Delta)$ from a node with "high" excess to the sink t or another node with "low" excess. This is in contrast to Fat-Path and most other algorithms, where these augmenting paths always terminate in the sink t. We allow other nodes as well in order to maintain $e_i \geq R_i$ are througout the algorithm. The purpose of these conditions is to make sure that we always stay "close" to a conservative labeling: recall Lemma 2.4 asserting that if (f, μ) is a Δ-feasible pair, then if we set the flow values to 0 on every non-tight arc, the resulting $\tilde{f}$ is a feasible solution to (P) not containing any flow-generating cycles. That is the reason why we need to call the cycle-canceling algorithm only once, at the initialization, in contrast to Fat-Path.

Similar ideas have been already used previously. The algorithm of Goldfarb, Jin and Orlin [11] also uses a single initial cycle-canceling and then performs highest-gain augmentations in a scaling framework, combined with a clever bookkeeping on the arcs. The algorithm in [34] does not perform any cycle cancellations and uses a homonymous notion of Δ-conservativeness that is closely related to ours; however, it uses a different problem setup (called "symmetric formulation"), and includes a condition stronger than $e_i \geq R_i$.

The way to the strongly polynomial bound

The basic principle of our strongly polynomial algorithm is motivated by Orlin's strongly polynomial algorithm for minimum cost circulations ([23], see also [2, Chapters 10.6-7]). The true purpose of the algorithm is to compute a dual optimal solution to (D). Provided a dual optimal solution, we can compute a primal optimal solution to (P) by a single maximum flow computation on the network of tight arcs (see Theorem 2.6(i)).

The main measure of progress is identifying an arc $ij \in E$ that must be tight in every dual optimal solution. Such an arc can be contracted, and an optimal dual solution to the contracted instance can be easily extended to an optimal dual solution on the original instance (see Sections 4.1, 5.1). The algorithm could be simply restarted from scratch in the contracted instance. Our algorithm Enhanced Continuous Scaling is somewhat more complicated and keeps the previous primal solution to achieve better running time bounds by a global analysis of all contraction phases.

We use a scaling-type algorithm to identify such arcs tight in every dual optimal solution. Our algorithm maintains a scaling parameter Δ, and a Δ-feasible pair (f, μ) such that $Ex^\mu(f) \leq 16m\Delta$. Using standard flow decomposition techniques, it can be shown that an arc ij with $f_{ij}^\mu \geq 17m\Delta$ must be positive in some optimal solution f^* to (P) (see Theorem 4.1). Then by primal-dual slackness it follows that this arc is tight in every dual optimal solution. Arcs with $f_{ij}^\mu \geq 17m\Delta$ will be called *abundant*.

A simple calculation (see the proof of Claim 5.4) shows that once $|b_i^\mu| \geq 32mn\Delta$ for a node $i \in V - t$, there must be an abundant arc leaving or entering i. Hence our goal is to design an algorithm where such a node appears within a strongly polynomial number of iterations.

A basic step in the scaling approaches (e.g. [9, 11, 34]) is sending Δ units of relabeled flow on a tight path; we shall call this a path augmentation. In all previous approaches, the scaling factor Δ remained fixed for a number of path augmentations, and reduces by a substantial amount (by at least a factor of two) for the next Δ-phase. Our main idea is what we call *continuous scaling*: the boundaries between Δ-phases are dissolved, and the scaling factor decreases continuously, even during the iterations that lead to finding the next path for augmentation. The precise description will be given in Section 2.5; in what follows, we give a high-level overview of some key features.

We shall have a set T_0 with nodes of "high" relabelled excess; another set N will be the set consisting of the sink t and further nodes with "low" relabelled excess. We will look for tight paths connecting a node in T_0 to one in N; we will send Δ units of relabeled flow along such a path. In an intermediate elementary step, we let T to denote the set of nodes reachable from T_0 on a tight path; if it does not intersect N, then we increase the labels μ_i for all $i \in T$ by the same factor α hoping that a new tight arc appears between T and $V \setminus T$, and thus T can be extended. We simultaneously decrease the value of Δ by the same factor α. Thus the relabeled excess of nodes in $V \setminus T$ increases relative to Δ. This might lead to changes in the sets T_0 and N; hence an elementary step does not necessarily terminate when a new tight arc appears, and therefore the value of α has to be carefully chosen.

This framework is undoubtedly more complicated than the traditional scaling algorithms. The main reason for this approach is the phenomenon one might call "inflation" in the previous scaling-type algorithms. There it might happen that the relabeling steps used for identifying the next augmenting paths increase some labels by very high amounts, and thus the relabeled flow remains small compared to Δ on every arc of the network - therefore a new abundant arc can never be identified. It could even be the case that most Δ-scaling phases do not perform any path augmentations at all, but only label updates: the relabeled excess at every node becomes smaller than Δ during the relabeling steps.[2]

The advantage of changing Δ continuously in our algorithm is that the ratios $|b_i^\mu|/\Delta$ are non-decreasing for every $i \in V - t$ during the entire algorithm. In the above described situation, these ratios are unchanged for $i \in T$ and increase for $i \in V \setminus T$. As remarked above, there must be an abundant arc incident to i once $|b_i^\mu|/\Delta \geq 32mn$.

We first present a simpler version of this algorithm, Continuous Scaling in Section 2.5, where we can only prove a weakly polynomial running time bound. Whereas the ratios $|b_i^\mu|/\Delta$ are nondecreasing, we are not able to prove that one of them eventually reaches the level $32mn$ in a strongly polynomial number of steps. This is since the set $V \setminus T$ where the ratio increases might always consist only of nodes where $|b_i^\mu|/\Delta$ is very small. The algorithm Enhanced Continuous Scaling in Section 4 therefore introduces one additional subroutine, called Filtration. In case $|b_i^\mu| < \Delta/n$ for every $i \in (V \setminus T) - t$, we "tidy-up" the flow inside $V \setminus T$, by performing a maximum flow computation here. This drastically reduces all relabeled excesses in $V \setminus T$, and thereby guarantees that most iterations of the algorithm will have to increase certain $|b_i^\mu|/\Delta$ values that are already at least $1/n$.

In summary, the strongly polynomiality of our algorithm is based on the following three main new ideas.

- The definition of Δ-feasible pairs, in particular, the condition on maintaining a security reserve

[2] However, to the extent of the author's knowledge, no actual examples are known for these phenomena in any of the algorithms.

R_i. It is a cleaner and more efficient framework than similar ones in [11] and [34]; we believe this is the “real” condition a scaling type algorithm has to maintain.

- Continuous scaling, that guarantees that the ratios $|b_i^\mu|/\Delta$ are nondecreasing during the algorithm. This is achieved by doing the exact opposite of [9, 11, 34] that use the natural analogue of the scaling technique for minimum cost circulations.
- The Filtration subroutine that intervenes in the algorithm whenever the nodes on a certain, relatively isolated part of the network have “unreasonably high” excesses as compared to the small node demands in this part.

2.5 The maximum flow subroutine

Standard maximum flow computation (see e.g. [2, Chapters 6-7]) will be a crucial subroutine in our algorithm. First and foremost, if an optimal labeling is provided, then an optimal solution to (P) can be obtained by computing a maximum flow. We now describe the subroutine Tight-Flow(S,μ), to perform such computations. In the weakly polynomial algorithm (Section 2.5), it will be used only twice: at the initialization and at the termination of the algorithm, in both cases with $S=V$. However, it will also be the key part of the subroutine Filtration in the strongly polynomial algorithm (Section 4), also applied to subsets $S\subsetneq V$.

The input of Tight-Flow(S,μ) is a node set $S\subseteq V$ with $t\in S$, and a labeling μ, that is a feasible solution to (D) when restricted to S. The subroutine returns a generalized flow f' supported on $E[S]$, such that μ restricted to S is a conservative labeling for f'. Let us define the arc set $\tilde{E}\subseteq E[S]$ as the set of tight arcs for μ:

$$\tilde{E} := \{ij\in E[S] : \gamma_{ij}^\mu = 1\}.$$

Let us extend S by a new source node s, and add an arc si from s to every $i\in S-t$; let $\tilde{E}'$ denote the union of $\tilde{E}$ and these new arcs. Let us set lower and upper arc capacities $\ell_{ij}:=0$, $u_{ij}:=\infty$ on all arcs of $\tilde{E}$; for $i\in S-t$, let $\ell_{si}:=-\infty$ and $u_{si}:=-b_i^\mu$.

Tight-Flow(S,μ) computes a maximum flow x from s to t on the network $(S\cup\{s\},\tilde{E}')$ with capacities ℓ and u. Let us define $f' : E[S]\to\mathbb{R}_{\geq 0}$ by $f'_{ij}:=x_{ij}\mu_i$ if $ij\in\tilde{E}$ and $f'_{ij}:=0$ otherwise. This completes the description of the subroutine Tight-Flow. Because of the possibly negative upper capacities on the si arcs, the maximum flow problem might be infeasible; in this case, the subroutine returns an error.

Theorem 2.6. *(i) If μ is an optimal solution to (D), then* Tight-flow(V,μ) *returns an optimal solution to (P).*

(ii) Assume that the maximum flow problem in Tight-flow(S,μ) *is feasible, and returns a vector f'. Then f' is a feasible solution to (P) on S, and*

$$e_i^\mu(f')\leq n\max_{j\in S-t}|b_j^\mu|\quad\forall i\in S.$$

(iii) Assume that the flow problem in Tight-flow(V,μ) *is feasible and returns a generalized flow f' with $Ex(f')<1/\bar{B}^3$. Then $Ex(f')=0$ must hold, that is, f' is an optimal solution to (P).*

Proof. To prove part *(i)*, assume μ is an optimal labeling. Let g be an optimal solution to (P). Let us define $x_{ij}:=g_{ij}^\mu$ if $ij\in E$ and $x_{si}:=\sum_{j:ij\in E}g_{ij}^\mu-\sum_{j:ji\in E}g_{ji}^\mu$ for every $i\in V-t$. By Theorem 2.3(i), $x_{si}=-b_i^\mu$ for all $i\in V-t$, and therefore x is a maximum flow, with $(\{s\},V)$ forming a minimum cut. Conversely, an arbitrary maximum flow must saturate every arc leaving s, and therefore we get

$e_i(f') = 0$ for every $i \in V - t$ for the f' returned by Tight Flow(V, μ). It is straightforward that all conditions in Theorem 2.3(i) are satisfied.

For part *(ii)*, first observe that if there is a feasible solution x to the flow problem, then $e_i(f') \geq 0$ must hold for every $i \in V - t$, due to the constraint $x_{si} \leq -b_i^\mu$; further, μ is a conservative labeling for f'. Let us pick a node $r \in S - t$ with $e_r(f') > 0$, and let $Z \subseteq S$ denote the set of nodes that can be reached from r on a directed path in the residual graph $(S, \tilde{E}_{f'})$, defined by

$$\tilde{E}_{f'} = \tilde{E} \cup \{ji : ij \in \tilde{E}, f'_{ij} > 0\}.$$

Note that $f'^{\mu}_{ij} = x_{ij}$ for every $ij \in \tilde{E}_{f'}$. Assume that $t \in Z$, that is, there is a directed path P from r to t in the residual graph. Since $e_r(f') > 0$, we have $x_{sr} < -b_i^\mu = u_{sr}$; hence sr and P give an augmenting path for the flow x, in a contradiction to its choice as a maximum flow.

We may thus conclude that $t \notin Z$. Hence $e_i^\mu(f') \geq 0$ for all $i \in Z$, and therefore

$$0 < e_r^\mu(f') \leq \sum_{i \in Z} e_i^\mu(f') = \sum_{i \in Z} \left(\sum_{j \in Z : ji \in \tilde{E}} x_{ji} - \sum_{j \in Z : ij \in \tilde{E}} x_{ij} - b_i^\mu \right) = -\sum_{i \in Z} b_i^\mu \leq n \max_{j \in S - t} |b_j^\mu|, \qquad (1)$$

proving part *(ii)* of the Theorem. Here we used that if $x_{ij} > 0$ then $i \in Z$ if and only if $j \in Z$.

Let us turn to part *(iii)*; assume that $e_r^\mu(f') > 0$ for some $r \in V - t$. The equation (1) can be further written as

$$0 < e_r^\mu(f') \leq \sum_{i \in Z} e_i^\mu(f') = -\sum_{i \in Z} b_i^\mu = -\frac{1}{\mu_r} \sum_{i \in Z} b_i \frac{\mu_r}{\mu_i}. \qquad (2)$$

For every $i \in Z$, there is a tight path P in $\tilde{E}_{f'}$ from r to i, that is, $\mu_r / \mu_i = \prod_{e \in P} 1/\gamma_e$. By our assumption on the encoding sizes, this product must be an integer multiple of $1/\bar{B}$. We further assumed that every b_i value is an integer multiple of $1/\bar{B}$. Hence every term $b_i \frac{\mu_r}{\mu_i}$ is an integer multiple of $1/\bar{B}^2$. Further, by $(\star)$, we have $rt \in E$, and $\gamma_{rt} \geq 1/\bar{B}$. By the conservativeness of μ w.r.t. to $\tilde{f}$, $\frac{1}{\mu_r} \geq \gamma_{rt} \geq 1/\bar{B}$. Consequently, the last expression in (2) must be at least $1/\bar{B}^3$ whenever it is nonzero. Therefore

$$1/\bar{B}^3 \leq \sum_{i \in Z} e_i^\mu(f') \leq Ex^\mu(f'),$$

contradicting our assumption. Hence it follows that $e_r(f) = 0$ for all $r \in V - t$. □

sectionThe Continuous Scaling algorithm

The algorithm Continuous Scaling is shown in Figure 1. The strongly polynomial algorithm Enhanced Continuous Scaling in Section 4 will be an improved variant of this. We shall always make assumptions $(\star)$, $(\star\star)$ and $(\star\star\star)$.

The algorithm starts with the subroutine Initialize, described in Section 2.6, that returns an initial flow f, along with a $\Delta = \bar{\Delta}$-conservative labeling μ such that $e_i^\mu < (d_i + 2)\Delta$ holds for every $i \in V$. This is based on the Maximum-mean-gain cycle-canceling algorithm as in [9, 25]. The main part of the algorithm (the while loop) consists of iterations. The value of the scaling parameter Δ is monotone decreasing, and all μ_i values are monotone increasing during the algorithm. In every iteration, a Δ-feasible pair (f, μ) is maintained. These iterations stop once the scaling parameter Δ decreases below $1/(17m\bar{B}^3)$. At this point we apply the subroutine Tight-flow(V, μ), as described in Section 2.5, to find an optimal solution by a single maximum flow computation.

The set N denotes the set consisting of t and all nodes with $e_i^\mu < (d_i + 1)\Delta$. The set T_0 consists of a certain set of nodes (but not all) with $e_i^\mu \geq (d_i + 2)\Delta$. The set T denotes a set of nodes that can be reached from T_0 on a tight path in the Δ-fat graph $E_f^\mu(\Delta)$. Both T_0 and T are initialized empty.

```
Algorithm Continuous Scaling
Initialize($V, E, b, \gamma, \bar{f}$) ;
$T_0 \leftarrow \emptyset$ ; $T \leftarrow \emptyset$ ;
While $\Delta \geq 1/(17m\bar{B}^3)$ do
    $N \leftarrow \{t\} \cup \{i \in V - t : e^\mu_i < (d_i + 1)\Delta\}$ ;
    if $N \cap T \neq \emptyset$ then
        pick $p \in T_0$, $q \in N \cap T$ connected by a tight path $P$ in $E^\mu_f(\Delta)$ ;
        send $\Delta$ units of relabeled flow from $p$ to $q$ along $P$ ;
        if $e^\mu_p < (d_p + 2)\Delta$ then $T_0 \leftarrow T_0 \setminus \{p\}$ ;
        $T \leftarrow T_0$ ;
    else
        if $\exists ij \in E^\mu_f(\Delta)$, $\gamma^\mu_{ij} = 1$, $i \in T$, $j \in V \setminus T$ then $T \leftarrow T \cup \{j\}$ ;
        else Elementary Step($T, T_0, f, \mu, \Delta$) ;
Tight-Flow($V, \mu$) ;
```

Figure 1: Description of the weakly polynomial algorithm

Every iteration first checks whether $N \cap T \neq \emptyset$. If yes, then nodes $p \in T_0$ and $q \in N \cap T$ are picked connected by a tight path P in the Δ-fat graph. Δ units of relabeled flow is sent from p to q on P: that is, f_{ij} is increased by $\Delta\mu_i$ for every $ij \in P$ (if ij was a reverse arc, this means decreasing f_{ji} by $\Delta\mu_j$). The only e_i values that change are e_p and e_q. If the new value is $e^\mu_p < (d_p + 2)\Delta$, then p is removed from T_0. The iteration finishes in this case by resetting $T = T_0$ (irrespective to whether p was removed or not).

Let us now turn to the case $N \cap T = \emptyset$. If there is a node $j \in V \setminus T$ connected by a tight arc in $E^\mu_f(\Delta)$ to T, then we extend T by j, and the iteration terminates. Otherwise, the subroutine Elementary Step(T, T_0, f, μ, Δ) is called. The precise description is given in Section 2.7; we give an outline below.

For a carefully chosen $\alpha > 1$, all μ_i values are multiplied by α for $i \in T$, and μ_i is left unchanged for $i \in V \setminus T$. At the same time, Δ is divided by α (this is the only step in the main part of the algorithm modifying the μ_i's and the value of Δ). The flow is divided by α on all non-tight arcs in $F^\mu[V \setminus T]$, and on every arc entering T. The value of α is chosen to be the largest such that the labeling remains Δ-feasible with the above changes, and further $e^\mu_i \leq 4(d_i + 2)\Delta$ holds for all $i \in V \setminus T$. If $\alpha = \infty$, then the algorithm terminates with an optimal solution. For finite α, all nodes i for which $e^\mu_i = 4(d_i + 2)\Delta$ holds after the change are added both to T_0 and to T. On the other hand, the e^μ_i values might also decrease both for $i \in T$ and $i \in V \setminus T$. If for some $i \in T_0$, the value of e^μ_i drops below $(d_i + 2)\Delta$, then i is removed from T_0, and T is reset to $T = T_0$. In every step when T_0 is not changed, a tight arc in $E^\mu_f(\Delta)$ leaving T must appear. Consequently, T will be extended in the next iteration. We shall prove the following running time bound:

Theorem 2.7. *The algorithm* Continuous Scaling *can be implemented to find an optimal solution for the uncapacitated formulation (P) in running time* $\max\{O(m(m+n\log n)\log \bar{B}), O(m^2 n \log^2 n)\}$.

The high level idea of the analysis is the following. The e^μ_i values for nodes $i \in T_0$ are non-increasing, and a path augmentation starting from i reduces e^μ_i by Δ. The node i leaves T_0 once e^μ_i drops below $(d_i + 2)\Delta$, and may enter again once it increases to $4(d_i + 2)\Delta$. As shown in Lemma 3.7, the value of Δ must decrease by at least a factor 2 between two such events. Also, it is easy to verify that within every $2n$ ELEMENTARY STEP operations, either a path augmentation must be carried out, or a node i must leave T_0 due to decrease in e_i^μ caused by label changes. These two facts together give a polynomial bound on the running time.

In the proof of Theorem 2.7, we outline a more efficient implementation of the algorithm, with all iterations between two path augmentations performed together.

For a problem in the standard form on n nodes, m arcs and complexity parameter B, the reduction in Section 7.1 shows that it can be transformed to an equivalent instance with $\le n+m$ nodes, $\le 2m$ arcs, and $\bar{B} \le 2B^{4m}$. Hence the theorem gives a running time $O(m^3 \log n \log B)$, assuming $n \le B$.

Our algorithm could be naturally adapted to work on a problem instance with both node demands and arc capacities; the reason for choosing the uncapacitated instance is its suitability for the strongly polynomial algorithm in Section 4. Such a modification would run in time $O(m^2(m+n\log n)\log B)$, matching the bound of Goldfarb et al. [13].

2.6 The Initialization subroutine

In this section we describe the INITIALIZE$(V, E, b, \gamma, \bar{f})$ subroutine. The input is a graph $G = (V, E)$, node demands $b_i : V \to \mathbb{R}$, gain factors $\gamma : E \to \mathbb{R}_{>0}$ and the initial generalized flow $\bar{f}$ guaranteed by the assumption $(\star\star)$. The initial value of $\Delta = \bar{\Delta}$ is computed and a Δ-feasible pair (f, μ) is returned such that $e_i^\mu < (d_i + 2)\Delta$ holds for every $i \in V - t$.

First, we use the MAXIMUM-MEAN-GAIN CYCLE-CANCELING algorithm by Radzik [25]. This paper uses the standard capacitated formulation (P_u) with finite capacities on the arcs. As a consequence, every flow-generating cycle can only generate a finite amount of flow. Our boundedness assumption $(\star\star\star)$, together with $(\star)$, guarantees the same property. Provided this boundedness property, Radzik's strongly polynomial bound extends verbatim to the uncapacitated formulation (P).

This returns a generalized flow g such that the residual graph E_g contains no flow generating cycles, that is, no cycles C with $\gamma(C) > 1$. Let us define $\mu_t := 1$ and for $i \in V - t$,

$$\mu_i := 1/\max\{\gamma(P) : P \subseteq E_g \text{ is a walk from } i \text{ to } t.\} \tag{3}$$

Such a path must exist according to assumption $(\star)$, and since $\gamma(C) \le 1$ for all cycles C, the walk giving the maximum can always be chosen to be a path. The μ_i values can be computed efficiently: note that they correspond to shortest paths with respect to the cost function $-\log\gamma_e$. To avoid computing logarithms, we may use a multiplicative version of Dijkstra's algorithm to obtain the μ_i values in strongly polynomial time.

After the cycle cancelling subroutine and computing the μ_i values, the subroutine TIGHT FLOW(V, μ) is called, as described in Section 2.5. This computes a generalized flow f'. We set $f = f'$, and set the initial $\Delta = \bar{\Delta} := \max_{i \in V-t} e_i^\mu$.

2.7 The Elementary step subroutine

Let (f, μ) be a Δ-feasible pair for $\Delta > 0$. Let $T \subseteq V$ be a (possibly empty) set of nodes with $e_i^\mu \le 4(d_i + 2)\Delta$ for every $i \in V$, with strict inequality whenever $i \in V \setminus T$. The subroutine (Figure 2) perfoms the following modifications for some $\alpha > 1$. The μ_i values are multiplied by α for $i \in T$, and left unchanged for $i \in V \setminus T$. The new value of the scaling parameter is set to Δ/α. Finally, the flow on non-tight arcs $ij \in F^\mu[V \setminus T]$ and on all arcs $ij \in E[V \setminus T, T]$ is divided by α.

The value of α is chosen maximal such that for the new values of f, μ, and Δ, (f, μ) is Δ-feasible, and further the modified excess $e_i \le 4(d_i + 2)\Delta\mu_i$ holds for every $i \in V$. For the latter, we need the

```
Subroutine ELEMENTARY STEP$(T, T_0, f, \mu, \Delta)$
$\alpha_1 \leftarrow \min\{\delta_i : i \in (V \setminus T) - t\}$, with $\delta_i$ as defined in (5) ;
$\alpha_2 \leftarrow \min\left\{\frac{1}{\gamma^\mu_{ij}} : ij \in E[T, V \setminus T]\right\}$ ;
$\alpha \leftarrow \min\{\alpha_1, \alpha_2\}$ ;
if $\alpha = \infty$ then
    set $f_{ti} = 0$ for all $ti \in E$: $\gamma^\mu_{ti} < 1$ ;
    return optimal flow $f$ and optimal relabeling $\mu$ ;
    TERMINATE.
$\Delta \leftarrow \frac{\Delta}{\alpha}$ ;
for $i \in T$ do $\mu_i \leftarrow \alpha\mu_i$ ;
for $ij \in F^\mu[V \setminus T] \cup E[V \setminus T, T]$ do $f_{ij} \leftarrow \frac{f_{ij}}{\alpha}$ ;
$T_0 \leftarrow T_0 \cup \{i : i \in V \setminus T,\ e^\mu_i = 4(d_i + 2)\Delta\}$ ;
$T \leftarrow T \cup T_0$ ;
if $\exists i \in T_0 : e^\mu_i < (d_i + 2)\Delta$ then
    $T_0 \leftarrow T_0 \setminus \{i : e^\mu_i < (d_i + 2)\Delta\}$ ;
    $T \leftarrow T_0$ ;
```

Figure 2: The Elementary Step subroutine

following definitions for every $i \in (V \setminus T) - t$. Let

$$\begin{array}{ll} F_1(i) := \delta^{in}(i) \cap F^\mu[V \setminus T], & r_1(i) := \sum_{j:ji \in F_1(i)} \gamma_{ji} f_{ji}, \\ F_2(i) := \delta^{in}(i) \setminus F_1(i), & r_2(i) := \sum_{j:ji \in F_2(i)} \gamma_{ji} f_{ji}, \\ F_3(i) := \delta^{out}(i) \cap (F^\mu[V \setminus T] \cup E[V \setminus T, T]), & r_3(i) := \sum_{j:ij \in F_3(i)} f_{ij}, \\ F_4(i) := \delta^{out}(i) \setminus F_3(i), & r_4(i) := \sum_{j:ij \in F_4(i)} f_{ij}. \end{array} \tag{4}$$

Note that $F_1(i)$ and $F_3(i)$ denote the set of those incoming and outgoing arcs where we wish to decrease the flow by a factor α. For every $i \in (V \setminus T) - t$, let us define

$$\delta_i := \frac{4(d_i + 2)\Delta\mu_i + r_3(i) - r_1(i)}{r_2(i) - r_4(i) - b_i}. \tag{5}$$

If the denominator is 0 then $\delta_i := \infty$ is set. We shall verify in the proof of Lemma 3.3 that the denominator is always nonnegative and the numerator is positive.

The subroutine (Figure 2) chooses α as the largest value subject to $\alpha \le \delta_i$ for all $i \in (V \setminus T) - t$, and $\alpha \le \frac{1}{\gamma^\mu_{ij}}$ for all arcs $ij \in E$ leaving the set T. If $\alpha = \infty$, then f becomes an optimal solution, after setting the value on all non-tight arcs leaving t to 0. If α is finite, the algorithm performs the above described modifications. Nodes i with $e^\mu_i = 4(d_i + 2)\Delta$ (that is, $\alpha = \delta_i$) are added to both T_0 and T. Finally, if e^μ_i drops below $(d_i + 2)\Delta$ for some $i \in T_0$, then all such nodes i will be removed from T_0, and T is reset to $T = T_0$. The validity of this subroutine is proved in Lemma 3.3.

3 Analysis of the Continuous Scaling algorithm

Lemma 3.1. *The subroutine* INITIALIZE$(V, E, b, \gamma, \bar{f})$ *returns a* Δ*-feasible pair* (f, μ) *with* $e^\mu_i \le (d_i + 2)\Delta$ *for every* $i \in V - t$*, and* $\Delta = \bar{\Delta} \le n\bar{B}^2$.

Proof. First, we have to verify that the flow problem in Tight-flow(V, μ) is feasible. We use the generalized flow g obtained by the Maximum-mean-gain cycle-canceling algorithm to verify this, by showing that μ is a conservative labeling for g. The nontrivial part is to prove $\gamma^\mu_{ij} \leq 1$ for every residual arc $ij \in E_g$.

Let $ij \in E_g$ be an arbitrary residual arc. Consider the $j - t$ path P^j with $\mu_j = 1/\gamma(P^j)$ in (3). Let P' denote the path resulting by adding the arc $ij \in E_g$ to the beginning of P^j. Then by definition, $1/\mu_i \geq \gamma(P') = \gamma_{ij}/\mu_j$, showing $\gamma^\mu_{ij} \leq 1$.

Let us now consider the maximum flow instance in Tight-flow(V, μ). Setting $x_{ij} := g^\mu_{ij}$ if $ij \in E$ and $x_{si} := \sum_{j:ij\in E} g^\mu_{ij} - \sum_{j:ji\in E} g^\mu_{ji}$ for every $i \in V - t$ gives a feasible solution. This guarantees the existence of the optimal solution f'.

It is straightforward by the construction that μ is a conservative labeling for f, and hence (f, μ) is Δ-feasible for arbitrary $\Delta > 0$. The condition $e^\mu_i \leq (d_i + 2)\Delta$ is also straightforward by definition.

Let us verify the bound on Δ. By Theorem 2.6(ii), we have $\Delta \leq n \max_{i\in V-t} |b_i|/\mu_i$. Our assumption on the encoding sizes give $|b_i| \leq \bar{B}$. Further, we have $1/\mu_i \leq \bar{B}$, according to the definition of $1/\mu_i = \gamma(P^i)$ for some $i-t$ path P^i, and the encoding assumptions on the γ_e values. □

The next straightforward claim justifies the path augmentation step carried out between $p \in T_0$ and $q \in N \cap T$ whenever $N \cap T \neq \emptyset$.

Claim 3.2. *Let (f, μ) be a Δ-feasible pair, and assume P is a tight path in $E^\mu_f(\Delta)$ from node p to node q, with $e^\mu_p(f) \geq \Delta + R^\mu_p$. Let us increase f_{ij} by $\Delta\mu_i$ if $ij \in P$ is a forward arc, and decrease f_{ji} by $\Delta\mu_j$ if $ij \in P$ is a backward arc; let f' denote the resulting flow. Then (f', μ) is also a Δ-feasible pair.*

We next prove some fundamental properties of the subroutine Elementary step, most importantly, that it maintains the Δ-feasibility of (f, μ). By induction, we may assume that the four conditions in the lemma always hold when Elementary step(T, T_0, f, μ, Δ) is called in the algorithm.

Lemma 3.3. *Let (f, μ) be a Δ-feasible pair for some $\Delta > 0$, and let $T \subseteq V - t$ satisfy the following conditions:*

- $e^\mu_i < 4(d_i + 2)\Delta$ *for all* $i \in (V \setminus T) - t$*;*
- $e^\mu_i \geq (d_i + 1)\Delta$ *for all* $i \in T$*;*
- $\gamma^\mu_{ij} < 1$ *for all* $ij \in E[T, V \setminus T]$*;*
- $f^\mu_{ij} \leq \Delta$ *for all* $ij \in E[V \setminus T, T]$.

Let f', μ', Δ', and e'_i denote the respective values at the end of Elementary step(T, T_0, f, μ, Δ)*. If $\alpha < \infty$, then the pair (f', μ') is Δ'-feasible. Further, the following statements hold.*

(i) $\alpha > 1$. If $\alpha = \infty$ then the modified flow returned by the algorithm is optimal to (P), and μ is optimal to (D).

(ii) $e'^{\mu'_i}_i \leq 4(d_i + 2)\Delta'$ for all $i \in V \setminus T$, and if $\alpha = \alpha_1$, then $\exists i \in V \setminus T$ such that equality holds.

(iii) $e'_i \leq e_i$ for all $i \in T$.

(iv) If $\alpha = \alpha_2$ then $\exists ij \in E$ with $i \in T$, $j \in V \setminus T$, and $\gamma^{\mu'}_{ij} = 1$.

Proof. For Δ'-feasibility, let us first verify $\gamma_{ij}^{\mu'} \le 1$ for all $ij \in E$. If $ij \in E[T]$ or $ij \in E[V \setminus T]$, then $\gamma_{ij}^{\mu'} = \gamma_{ij}^{\mu}$. If $ij \in E[T, V \setminus T]$, then we have $\gamma_{ij}^{\mu'} = \alpha\gamma_{ij}^{\mu} \le 1$ due to the choice $\alpha \le \alpha_2$. Finally, if $ij \in E[V \setminus T, T]$, then $\gamma_{ij}^{\mu'} = \gamma_{ij}^{\mu}/\alpha < 1$. The next two claims verify the remaining properties needed for Δ'-feasibility.

Claim 3.4. *If $\gamma_{ij}^{\mu'} < 1$ for an arc $ij \in E$, then ${f'_{ij}}^{\mu'} = f_{ij}^{\mu}/\alpha \le \Delta/\alpha = \Delta'$.*

Proof. Let us first assume $i \in T$; the first equality follows by $f'_{ij} = f_{ij}$, $\mu'_i = \mu_i\alpha$. The inequality $f_{ij}^{\mu} \le \Delta$ is due to the Δ-feasibility of f, because of $\gamma_{ij}^{\mu} < 1$. If $j \in V \setminus T$, this is included among the assumptions, whereas if $j \in T$, then it follows by $\gamma_{ij}^{\mu} = \gamma_{ij}^{\mu'} < 1$.

Consider now the case $i \in V \setminus T$. If also $j \in V \setminus T$, then $\gamma_{ij}^{\mu} = \gamma_{ij}^{\mu'} < 1$, and hence $f'_{ij} = f_{ij}/\alpha$, as we decrease the flow values by a factor α on arcs $F^{\mu}[V \setminus T]$; the inequality $f_{ij}^{\mu} \le \Delta$ follows again by the Δ-feasibility of f. If $j \in T$, that is, $ij \in E[V \setminus T, T]$, then we must again have $f'_{ij} = f_{ij}/\alpha$, and $f_{ij}^{\mu} \le \Delta$ is included among the assumptions. □

Claim 3.5. *The inequality $e'_i \ge R'_i$ holds for all $i \in V - t$, where R'_i denotes the f' flow entering i on non-tight arcs for μ'.*

Proof. We have $e_i \ge R_i$ by the Δ-feasibility of f.

Case I: $i \in V \setminus T$**.** Since $f' \le f$, the change of flow on outgoing arcs may only increase e_i. If $f'_{ji} < f_{ji}$ on an incoming arc $ji \in E$, then $j \in V \setminus T$ must hold. Therefore $\gamma_{ji}^{\mu'} = \gamma_{ji}^{\mu}$, and hence ji must be a non-tight arc for both μ and μ'. The change on ji decreases e_i by $(1 - 1/\alpha)\gamma_{ji}f_{ji}$, and causes the same change in the value of R_i.

Case II: $i \in T$**.** By the assumption of the lemma, $e_i^{\mu} \ge (d_i + 1)\Delta$. The flow on outgoing arcs is unchanged. Let $ji \in E$ be an incoming arc with $f'_{ji} < f_{ji}$. We must have $j \in V \setminus T$ and thus $f_{ji} \le \Delta\mu_j$ by assumption; further, $\gamma_{ji}^{\mu} \le 1$ by the Δ-feasibility of (f, μ). Hence it follows that $\gamma_{ji}f_{ji} < \Delta\gamma_{ji}\mu_j \le \Delta\mu_i$. This enables us to bound the value e'_i. Let λ denote the number of arcs ji with $j \in V \setminus T$. Using also the assumption $e_i \ge (d_i + 1)\Delta\mu_i$, we have

$$\begin{aligned} e'_i = e_i - \sum_{j \in V \setminus T : ji \in E} (\gamma_{ji}f_{ji} - \gamma_{ji}f'_{ji}) &\ge e_i - \sum_{j \in V \setminus T : ji \in E} (\Delta\mu_i - \gamma_{ji}f'_{ji}) \\ \ge \sum_{j \in V \setminus T : ji \in E} \gamma_{ji}f'_{ji} + (d_i + 1 - \lambda)\Delta\mu_i > R'_i. \end{aligned}$$

In the last inequality, we use that if ji is a non-tight arc with $j \in T$, then $\gamma_{ji}f'_{ji} \le \Delta'\mu'_i = \Delta\mu_i$, and that the total number of such arcs is $\le d_i - \lambda$. □

Let us now verify claims *(i)-(iv)*. We first show that in the formula (5) defining δ_i, the denominator is nonnegative and $\delta_i > 1$. Note that

$$r_1(i) + r_2(i) - r_3(i) - r_4(i) - b_i = e_i \ge R_i \ge r_1(i). \tag{6}$$

The equality is by the definition of the four terms; the first inequality is required by Δ-feasibility, and the second since the definition of R_i includes all terms in $r_1(i)$. This shows that the denominator is $r_2(i) - r_4(i) - b_i \ge r_3(i) \ge 0$. The inequality $\delta_i > 1$ then follows by the equality in (6) and the assumption $4(d_i + 2)\Delta\mu_i > e_i$.

For *(i)*, the above argument gives $\alpha_1 > 1$. It is easy to see that $\alpha_2 > 1$, and hence $\alpha > 1$ follows. For the second part, let us analyze the $\alpha = \infty$ case. First, we show that $T = \emptyset$ must hold. For a contradiction, assume $T \ne \emptyset$. We have $t \in V \setminus T$ is assumed, and every $j \in V - t$ is connected by an arc to t by ($\star$). Therefore the set of arcs defining α_2 is always nonempty, showing that α must be finite.

We thus have $T = \emptyset$. Since $\alpha_1 = \infty$, we must have $\delta_i = \infty$ for every $i \in V - t$, that is, the denominator in (5) is always 0. According to (6), this is only possible if $r_3(i) = 0$ for every $i \in V - t$. This means that for every $ij \in F^\mu$, if $f_{ij} > 0$, then $i = t$ must hold. As a further consequence of (6), we have $e_i = R_i = r_1(i)$ for every $i \in V - t$. Combining these two, for every $i \in V - t$ we obtain $e_i = r_1(i) = \gamma_{ti} f_{ti}$ if $ti \in F^\mu$, and $e_i = r_1(i) = 0$ if $ti \notin F^\mu$. After the algorithm sets the value of all these arcs to 0, μ becomes a conservative labeling, and $e_i = 0$ for all $i \in V - t$, yielding primal and dual optimality according to Theorem 2.3(i).

Let us now prove claim *(ii)*. The flow on the arcs incident to i is divided by α on all arcs in $F_1(i)$ and $F_3(i)$, and left unchanged on arcs in $F_2(i)$ and $F_4(i)$. Therefore, $e_i'^{\mu_i'} \le 4(d_i + 2)\Delta'$ follows whenever $\alpha \le \delta_i$. The claims on nodes/arcs with equalities in *(ii)* and *(iv)* are straightforward. Finally, *(iii)* follows since if $i \in T$, then the flow is unchanged on outgoing arcs and on arcs incoming from T, but decreases on arcs incoming from $V \setminus T$. □

3.1 Bounding the number of iterations

Let $\Delta^{(\tau)}$ denote the value of the scaling factor at the beginning of the τ'th iteration; clearly, $\Delta^{(1)} \ge \Delta^{(2)} \ge \ldots \ge \Delta^{(\tau)}$. Let $f^{(\tau)}, \mu^{(\tau)}, e^{(\tau)}$ and $T^{(\tau)}$ denote the respective vectors and set T at the beginning of iteration τ.

Let us classify the iterations into three categories. The iteration θ is *shrinking*, if $T^{(\theta)} \setminus T^{(\theta+1)} \neq \emptyset$. Every iteration with a path augmentation is shrinking, since T is reset to T_0, although it contained some other nodes, in particular, the endpoint q of the path previously. The other type of shrinking iteration is when Elementary step is performed, and for some $i \in T_0$, the value of e_i^μ is decreased below $(d_i + 2)\Delta$.

The iteration θ is *expanding*, if $T^{(\theta)} \subsetneq T^{(\theta+1)}$. This can either happen if the iteration only consists of extending T by adding a new node reachable by a tight arc in the Δ-fat graph, or if T_0 is extended in Elementary step, and no node is removed from T_0. An iteration that is neither shrinking nor expanding is called *neutral.* Note that in a neutral iteration we must perform Elementary step, and further we must have $T^{(\theta)} = T^{(\theta+1)}$. We claim that the iteration following the neutral iteration θ must be either expanding or shrinking. Indeed, if $T^{(\theta+1)} \cap N^{(\theta+1)} \neq \emptyset$, then it will be shrinking. Otherwise, Lemma 3.3(iv) guarantees that it must be expanding. The main goal of this section is to prove the following lemma.

Lemma 3.6. *For the starting value $\Delta^{(1)} = \bar{\Delta}$ and arbitrary integer $\tau \ge 1$, we have*

$$\tau \le 26mn \log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}}.$$

Further, the total number of shrinking iterations among the first τ is at most

$$13m \log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}}.$$

An important quantity in our analysis will be

$$\beta_i := \frac{e_i}{\Delta \mu_i};$$

let $\beta_i^{(\tau)}$ denote the corresponding value at the beginning of iteration τ. Let $\alpha^{(\tau)}$ denote the value of α in iteration τ if the subroutine Elementary Step is called, and let $\alpha^{(\tau)} = 1$ otherwise. Note that the value of the scaling factor only changes in the subroutine Elementary Step. Therefore

$$\frac{\bar{\Delta}}{\Delta^{(\tau+1)}} = \prod_{\theta\in[1,\tau]} \alpha^{(\theta)} \quad \forall \tau \in \mathbb{Z},\ \tau > 1.$$

Lemma 3.7. *During the first τ iterations, a node i may enter the set T_0 altogether at most $\log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}}$ times.*

Before proving the lemma, let us show how it can be used to bound the number of iterations.

Proof of Lemma 3.6. Let us consider the potential

$$\Psi := \sum_{i\in T_0} \lfloor \beta_i - (d_i+1) \rfloor. \tag{7}$$

Initially, $T_0 = \emptyset$ and therefore $\Psi = 0$. Note that every term is positive in every step of the algorithm, since nodes with $\beta_i < (d_i + 2)$ are immediately removed from T_0. The subroutine Elementary step may only decrease the value of Ψ: Lemma 3.3(iii) guarantees that if $i \in T_0$, then β_i may only decrease during the subroutine, since $e'_i \le e_i$ and $\Delta'\mu'_i = \Delta\mu_i$.

Every shrinking iteration must decrease Ψ by at least one. Indeed, a path augmentation decreases e_p by $\Delta\mu_p$ for the starting node p, which decreases $\lfloor \beta_p - (d_p + 1) \rfloor$ by one. No other β_i value is modified for $i \in T_0$. Next, consider the case when a shrinking iteration removes some nodes i from T_0 after performing Elementary step because of $\beta_i < (d_i + 2)$. In the previous iteration, we must have had $\beta_i \ge (d_i + 2)$ for such nodes, hence Ψ decreases by at least 1.

When a node i enters T_0, then it increases Ψ by $(3d_i + 7)$. Assume that the node i enters T_0 altogether λ_i times between iterations 1 and τ. Then Lemma 3.7 gives $\lambda_i \le \log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}}$. Therefore the total increase in the Ψ value between iterations 1 and τ is bounded by

$$\sum_{i\in V-t} (3d_i+7)\lambda_i \le \sum_{i\in V-t} (3d_i+7)\log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}} \le (6m+7n)\log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}} \le 13m \log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}}$$

This bounds the number of shrinking iterations (recall the assumption $n \le m$). Between two subsequent shrinking iterations, all phases are expanding or neutral. Every expanding iteration increases T, and every neutral iteration is followed by a shrinking or an expanding iteration. Therefore the total number of iterations between two subsequent shrinking iterations is $\le 2n$, giving an overall bound

$$26mn \log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}}$$

on the number of iterations. □

The proof of Lemma 3.7 is based on the following simple claim.

Claim 3.8. *Let β'_i denote the new value of β_i after performing the subroutine* Elementary Step(T, T_0, f, μ, Δ)*, that computes the value α. For every node $i \in V - t$, we have*

$$\beta'_i \le \alpha^2 \max\{\beta_i, d_i\}.$$

Proof. Let Δ and $\Delta' = \Delta/\alpha$ denote the scaling factor before and after performing the subroutine Elementary Step(T, T_0, f, μ, Δ). If $i \in T$, then $e'_i \le e_i$ by Lemma 3.3(iii) and $\Delta'\mu'_i = \Delta\mu_i$, and hence $\beta'_i \le \beta_i$, implying the claim. Assume therefore that $i \in V \setminus T$. We have $f' \le f$, and the flow changes on arcs entering i may only decrease e_i. Recall that $F_3(i)$ denotes the set of outgoing arcs ij where $f'_{ij} < f_{ij}$. Note that $f_{ij} \le \Delta\mu_i$ on every such arc. We get the upper bound

$$e'_i \le e_i + \sum_{j:ij\in F_3(i)} (1-1/\alpha) f_{ij} \le e_i + (1-1/\alpha)|F_3(i)|\Delta\mu_i \le e_i + (\alpha-1)|F_3(i)|\Delta\mu_i.$$

In the last inequality, we used $1-1/\alpha \le \alpha - 1$, which is true for every $\alpha > 0$. Using further that $\Delta'\mu'_i = \Delta\mu_i/\alpha$, we get

$$\beta'_i = \frac{e'_i}{\Delta'\mu'_i} \le \frac{\alpha(e_i + (\alpha-1)|F_3(i)|\Delta\mu_i)}{\Delta\mu_i} = \alpha\frac{e_i}{\Delta\mu_i} + (\alpha-1)|F_3(i)| \le$$
$$\alpha\beta_i + (\alpha-1)d_i \le (2\alpha-1)\max\{\beta_i, d_i\} \le \alpha^2 \max\{\beta_i, d_i\},$$

completing the proof. □

Proof of Lemma 3.7. Let $\tau_1 < \tau_2 < \ldots < \tau_\lambda \le \tau$ denote the iterations when i enters T_0 up to iteration τ. This means that $\beta_i^{(\tau_\ell+1)} = 4(d_i+2)$ for $1 \le \ell \le \lambda$.

For $1 \le \ell \le \lambda$, let us define τ'_ℓ to be the largest value $\tau'_\ell \le \tau_\ell$ such that $\beta_i^{(\tau'_\ell)} < (d_i+2)$. Note that these values must exist and satisfy $\tau_{\ell-1} < \tau'_\ell \le \tau_\ell$ for $\ell > 1$. Indeed, for $\ell = 1$, we assumed that at the beginning of the algorithm $\beta_i^{(1)} < (d_i+2)$. For $\ell > 1$, note that i must leave T_0 in some iteration θ between $\tau_{\ell-1}$ and τ_ℓ, and this can happen only if $\beta_i^{(\theta)} < (d_i+2)$.

In iteration τ'_ℓ, we have $i \notin T_0$, since once the excess e_i drops below $(d_i+2)\Delta\mu_i$, the node i is immediately removed from T_0. By definition, i will be added to T_0 in iteration τ_ℓ.

The e_i values may change in two ways between iterations τ'_ℓ and τ_ℓ: either during a path augmentation or in the subroutine Elementary step. We claim that no path augmentation changes e_i in the iterations $\tau'_\ell \le \theta \le \tau_\ell$. Indeed, the only values that change are at the starting point p and endpoints q of the tight path P. We cannot have $i = p$ as $i \notin T_0$ during these iterations. Assume now $i = q$ is the endpoint; therefore $e_i^{(\theta)} < (d_i+1)\Delta^{(\theta)}\mu_i^{(\theta)}$. This clearly cannot be the case for $\tau'_\ell < \theta \le \tau_\ell$ by the maximal choice of τ'_ℓ. Let us consider the case $\theta = \tau'_\ell$. The path augmentation terminating in $i = q$ increases $e_i^{(\tau'_\ell)}$ by $\Delta^{(\tau'_\ell)}\mu_i^{(\tau'_\ell)}$. However, we had $e_i^{(\tau'_\ell)} < (d_i+1)\Delta^{(\tau'_\ell)}\mu_i^{(\tau'_\ell)}$, and therefore

$$e_i^{(\tau'_\ell+1)} = e_i^{(\tau'_\ell)} + \Delta^{(\tau'_\ell)}\mu_i^{(\tau'_\ell)} < (d_i+2)\Delta^{(\tau'_\ell+1)}\mu_i^{(\tau'_\ell+1)},$$

again a contradiction to the choice of τ'_ℓ. (Note that if a path augmentation is done in iteration τ'_ℓ, then the values of Δ and μ do not change).

Hence all changes in the value of e_i are due to modifications in Elementary step. Consequently,

$$4 = \frac{4(d_i+2)}{(d_i+2)} < \frac{\beta^{(\tau_\ell+1)}}{\max\{\beta^{(\tau'_\ell)}, d_i\}} \le \frac{\beta^{(\tau'_\ell+1)}}{\max\{\beta^{(\tau'_\ell)}, d_i\}} \prod_{\theta\in[\tau'_\ell+1,\tau_\ell]} \frac{\beta^{(\theta+1)}}{\beta^{(\theta)}} \tag{8}$$

For $\theta \in [\tau'_\ell+1, \tau_\ell]$, we assumed $\beta^{(\theta)} > d_i$, and hence Claim 3.8 gives that $\frac{\beta^{(\theta+1)}}{\beta^{(\theta)}} \le \left(\alpha^{(\theta)}\right)^2$. The same claim bounds the first term by $\le \left(\alpha^{(\tau'_\ell)}\right)^2$. Hence we get

$$4 \le \left(\prod_{\theta\in[\tau'_\ell,\tau_\ell]} \alpha^{(\theta)}\right)^2.$$

Adding the logarithms of these inequalities for all $\ell = 1, \dots, \lambda$, we obtain

$$\lambda \le \sum_{\theta \in [1,\tau]} \log_2 \alpha^{(\theta)} = \log_2 \frac{\bar{\Delta}}{\Delta^{(\tau+1)}},$$

completing the proof. □

3.2 The termination of the algorithm

The algorithm either terminates in Elementary step or by the final subroutine Tight-flow(V, μ). Optimality for the first case was already proved in Lemma 3.3(i). The next claim addresses the second case.

Lemma 3.9. *The final f' and μ returned by the subroutine* Tight-flow(V, μ) *are a primal and a dual optimal solution to (P) and (D), respectively.*

Proof. We show that the flow problem in Tight-flow(V, μ) is feasible and $Ex^\mu(f') < 1/\bar{B}^3$. Then optimality follows by Theorem 2.6(iii). At the termination of the While iterations of the algorithm Continuous Scaling, we have

$$Ex^\mu(f) = \sum_{i \in V-t} e_i^\mu \le 4\Delta \sum_{i \in V} (d_i + 2) = (8m + 8n)\Delta.$$

Let us define $\tilde{f}$ by $\tilde{f}_{ij} = 0$ if $ij \in F^\mu$ and $\tilde{f}_{ij} = f_{ij}$ otherwise. By Lemma 2.4,

$$Ex^\mu(\tilde{f}) < Ex^\mu(f) + |F^\mu|\Delta \le (9m + 8n)\Delta < 1/\bar{B}^3,$$

since $\Delta < 1/(17m\bar{B}^3)$ at the termination. The proof is complete by verifying the feasibility of the flow problem and showing that $Ex^\mu(f') \le Ex^\mu(\tilde{f})$.

Let us define the feasible solution $\tilde{x}$ to the flow problem in Tight Flow as follows. We use the notation introduced in the description of the subroutine in Section 2.5. Let $\tilde{x}_{ij} := \tilde{f}_{ij}^\mu$ for $ij \in E$. Further, for $i \in V - t$, let us set $\tilde{x}_{si} := \sum_{j:ij \in E} \tilde{f}_{ij}^\mu - \sum_{j:ji \in E} \tilde{f}_{ji}^\mu$. The conservativeness of $\tilde{f}$ implies that $\tilde{x}_{si} \le -b_i^\mu = u_{si}$. Therefore $\tilde{x}$ is a feasible solution to the flow problem. The value of this flow $\tilde{x}$ (i.e. the sum of the flow on the arcs leaving s) is

$$\sum_{i \in \tilde{V}-t} \tilde{x}_{si} = -\sum_{i \in V-t} (b_i^\mu + e_i^\mu(\tilde{f})) = -Ex^\mu(\tilde{f}) - \sum_{i \in V-t} b_i^\mu.$$

Similarly, the value of the flow x found by Tight Flow is $-Ex^\mu(f') - \sum_{i \in V-t} b_i^\mu$. Since x is maximal, it follows that $Ex^\mu(f') \le Ex^\mu(\tilde{f})$. □

3.3 Running time analysis

Proof of Theorem 2.7. The starting value of the scaling factor is $\bar{\Delta} \le n\bar{B}^2$ by Lemma 3.1, and we terminate once $\Delta^{(\tau+1)} < 1/(17m\bar{B}^3)$. Therefore $\log \frac{\bar{\Delta}}{\Delta^{(\tau+1)}} \in O(\log \bar{B})$ (we may assume $\log \bar{B}$ is larger than m). According to Lemma 3.6, the number of iterations of the algorithm is $O(mn \log \bar{B})$, out of them $O(m \log \bar{B})$ shrinking ones. We have to execute two maximum flow computations, that can be done in $O(nm)$ time using the recent algorithm by Orlin [24]. The initial cycle canceling subroutine can be executed in time $O(m^2 n \log^2 n)$, see Radzik [25]. The proof is complete by showing that the part of the algorithm between two shrinking iterations can be implemented in $O(m + n \log n)$ time.

We implement all these iterations together via a Dijkstra-type algorithm, using the Fibonacci-heap data structure [7], see also [2, Chapter 4.7]. The precise details are given in Section 6, see Figure 5; here we outline the main ideas only. Each label is modified only once, at the beginning of the subsequent shrinking iteration; for every i, it is sufficient to record the value of α at the moment when i enters T. We have to modify the f_{ij} values accordingly. We maintain a heap with elements $i \in V \setminus T$, with five keys associated to each of them. The main key for $i \in V \setminus T$ corresponds to the minimum of the $1/\gamma^{\mu}_{ji}$'s for $j \in T$, and of δ_i. The four auxiliary keys store the flow values $r_1(i), \dots, r_4(i)$, as in the definition (5) of δ_i. We choose the next i who enters T with the minimal main key. If the minimal key corresponds to the δ_i value, then i enters both T and T_0; otherwise, it enters only T. We remove i from the heap, and update the keys on the adjacent nodes. We maintain another heap structure on T to identify events when for a node $i \in T_0$, $e^{\mu}_i < (d_i + 2)\Delta$ happens, or when a node in $T \setminus T_0$ enters N.

Overall, these modifications entail $O(m)$ key modifications only; the keys can be initialized in total time $O(m)$. We therefore obtain the running time $O(m+n \log n)$ as for Dijkstra's algorithm. $\square$

4 The strongly polynomial algorithm

The while loop of the algorithm Enhanced Continuous Scaling proceeds very similarly to Continuous Scaling, with the addition of the special subroutine Filtration, described in Section 4.2. However, the termination criterion is quite different. As discussed in Section 2.4, the goal is to find a node $i \in V - t$ with $\frac{|b^{\mu}_i|}{\Delta} \geq 32mn$. There must be an abundant arc incident to such a node that we can contract and continue the algorithm in the smaller graph. Section 4.1 describes the abundant arcs and the contraction operation.

Let us now give some motivation for the algorithm; we focus on the sequence of iterations leading to the first abundant arc. Consider the set

$$D := \left\{ i \in V - t : \frac{|b^{\mu}_i|}{\Delta} \geq \frac{1}{n} \right\}.$$

Our aim is to guarantee that most iterations when Δ is multiplied by α will multiply $\frac{|b^{\mu}_i|}{\Delta}$ by α for some $i \in D$. This will ensure that $\frac{|b^{\mu}_i|}{\Delta} \geq 32mn$ happens within $O(nm \log n)$ number of steps. Note that in the subroutine Elementary step(T, f, μ, Δ), the $\frac{|b^{\mu}_i|}{\Delta}$ ratio is multiplied by α for all nodes $i \in V \setminus T$ and remains unchanged for $i \in T$.

Therefore we modify the while loop of Continuous Scaling as follows. If $(V \setminus T) \cap D \neq \emptyset$, Elementary step$(T, f, \mu, \Delta)$ is performed identically. If $(V \setminus T) \cap D = \emptyset$, then before Elementary step(T, f, μ, Δ), the special subroutine Filtration$(V \setminus T, f, \mu)$ is executed, performing the following changes.

The value of f is set to 0 for every arc entering T, and f_{ij} is left unchanged for $i \in T$. The flow value on arcs inside $E[V \setminus T]$ is replaced by an entirely new flow f' computed by Tight Flow$(V \setminus T, \mu)$.

An important part of the analysis is Theorem 2.6(ii), asserting that $e^{\mu}_i(f') \leq n \max_{j \in (V \setminus T) - t} |b^{\mu}_j|$. This will imply that either the set D must be extended in the iteration following Filtration$(V \setminus T, f, \mu)$, or there must be a shrinking one among the next two iterations (Lemma 5.11(ii)). Note that once a node enters D, it stays there until the next contraction.

4.1 Abundant arcs and contractions

Given a Δ-feasible pair (f, μ), we say that an arc $pq \in E$ is *abundant*, if $f^{\mu}_{pq} \geq 17m\Delta$. The importance of abundant arcs is that they must be tight in all dual optimal solutions. This is a corollary of the following theorem.

Theorem 4.1. *Let (f,μ) be a Δ-feasible pair. Then there exists an optimal solution f^* such that*

$$||f^\mu - f^{*\mu}||_\infty \le Ex^\mu(f) + (|F^\mu| + 1)\Delta.$$

The standard proof using flow decompositions is given in the Appendix; it can also be derived from Lemma 5 in Radzik [26]. For the flow f in an iteration with scaling factor Δ, we have $Ex^\mu(f) \le \sum_{i\in V-t} 4(d_i+2)\Delta < (8m+8n-8)\Delta \le (16m-8)\Delta$. Further, $|F^\mu| \le m$. This gives the following corollary; the last part follows by primal-dual slackness conditions.

Corollary 4.2. *Let (f,μ) be the Δ-feasible pair during the algorithm. If for an arc $pq \in E$, $f^\mu_{pq} \ge 17m\Delta$, then $f^*_{pq} > 0$ for some optimal solution f^* to (P). Consequently, $\gamma_{pq}\mu^*_p = \mu^*_q$ for every optimal solution μ^* to (D).*

Once an abundant arc pq is identified in the Enhanced Continuous Scaling algorithm, it is possible to reduce the problem by contracting pq. Consider the problem instance (V,E,t,b,γ). The *contraction of the arc pq* returns a problem instance (V',E',t',b',γ') with $t' := t$, as follows.

Case I: $p \ne t$. Let $V' = V \setminus \{p\}$, and add an arc $ij \in E'$ if $ij \in E$ and $i,j \ne p$. For every arc $ip \in E$, add an arc $iq \in E'$, and for every arc $pi \in E$, $i \ne q$, add an arc $qi \in E'$. Set the gain factors as $\gamma'_{ij} := \gamma_{ij}$ if $i,j \ne p$, $\gamma'_{iq} := \gamma_{ip}\gamma_{pq}$ and $\gamma'_{qi} := \gamma_{pi}/\gamma_{pq}$. Let us set $b'_i := b_i$ if $i \ne q$, and $b'_q := b_q + \gamma_{pq}b_p$.

Case II: $p = t$. Let $V' = V \setminus \{q\}$, and add an arc $ij \in E'$ if $ij \in E$ and $i,j \ne q$. For every arc $iq \in E$, $i \ne p$, add an arc $ip \in E'$, and for every arc $qi \in E$, add an arc $pi \in E'$. Set the gain factors as $\gamma'_{ij} := \gamma_{ij}$ if $i,j \ne p$, $\gamma'_{ip} := \gamma_{iq}/\gamma_{pq}$ and $\gamma'_{pi} := \gamma_{qi}\gamma_{pq}$. Let us set $b'_i := b_i$ if $i \ne p$, and $b'_p := b_p + b_q/\gamma_{pq}$.

In both cases, if parallel arcs are created, keep only one that maximizes the γ' value. Let $s := q$ in the first and $s := p$ in the second case. If a loop incident to s is created (corresponding to a qp arc), remove it.

Assume further we are given a generalized flow f and a labeling μ with $\gamma^\mu_{pq} = 1$ in the instance. We define the image labels μ', by simply setting $\mu'_i = \mu_i$ for all $i \in V'$ in both cases. Note that we will have ${b'_s}^{\mu'} = b^\mu_p + b^\mu_q$ in both cases.

As for the generalized flow, let $f'_{ij} := f_{ij}$ whenever $i,j \ne s$. For every $i \in V' \setminus \{s\}$, we let $f'_{is} := f_{ip} + f_{iq}$. Further, in Case I, we let $f'_{si} := \gamma_{pq}f_{pi} + f_{qi}$, whereas in Case II, we let $f'_{si} := f_{pi} + f_{qi}/\gamma_{pq}$. If one of these arcs is not in E, then we substitute the corresponding value by 0. Recall that in the construction, we keep the larger gain factor from two parallel incoming or outgoing arcs.

The above transformation of an instance, generalized flow and labels will be executed by the subroutine Contract(pq). Note that if the original instance satisfies ($\star$), ($\star\star$), and ($\star\star\star$), then these also hold for the contracted instance; the contracted image of the initial feasible solution $\bar{f}$ is feasible for the contracted instance.

Let us also describe the reverse operation, Reverse(pq), that transforms a dual solution on the contracted instance to a dual solution in the original one. Assume μ' is a dual solution in the graph obtained by the contraction of pq. Let us set $\mu_i := \mu'_i$ for all $i \in V - s$. In the first case ($p \ne t$, $s = q$), let us set $\mu_p := \mu'_q/\gamma_{pq}$, whereas in the second case ($p = t$, $s = p$), let us set $\mu_q := \mu'_p\gamma_{pq} = \gamma_{pq}$.

4.2 The Filtration subroutine

A typical iteration of the Enhanced Continuous Scaling algorithm (Figure 4) will be the same as in Continuous Scaling, with adding one additional subroutine, Filtration($V \setminus T, f, \mu$) before performing Elementary step(T, T_0, f, μ, Δ). This subroutine is executed if $|b_i^\mu| < \Delta/(16^k n)$ holds for all $i \in (V \setminus T) - t$, where k is the number of arcs contracted so far, initially $k = 0$.

Filtration$(V \setminus T, f, \mu)$ (Figure 3) performs the subroutine Tight Flow$(V \setminus T, \mu)$, as described in Section 2.5. This replaces f by an entirely new flow f' on the arcs in $E[V \setminus T]$. We further set $f_{ij} = 0$ on all arcs entering T, and keep the original f value on all other arcs (that is, arcs in $E[T] \cup E[T, V \setminus T]$).

```
Subroutine Filtration($V \setminus T, f, \mu$)
$f' \leftarrow$ Tight Flow($V \setminus T, \mu$) ;
for $ij \in E$ do
    if $ij \in E[V \setminus T]$ then $f_{ij} \leftarrow f'_{ij}$ ;
    if $ij \in E[V \setminus T, T]$ then $f_{ij} \leftarrow 0$ ;
```

Figure 3: The Filtration subroutine

4.3 The Enhanced Continuous Scaling Algorithm

We are ready to describe our strongly polynomial algorithm, shown on Figure 4. The algorithm consists of iterations similar to Continuous Scaling, with the addition of the above described Filtration subroutine. This subroutine might decrease e_i^μ values below $(d_i + 2)\Delta$ for some $i \in T_0$; also, $e_i^\mu < (d_i + 1)\Delta$ might happen for some $i \in T$, that is, i is added to the set $N \cap T$. If either of these events happen, we proceed to the next iteration without performing the subroutine Elementary step(T, T_0, f, μ, Δ). Further, if there are nodes $i \in T_0$ where the e_i^μ values drop below $(d_i + 2)\Delta$, then we remove all such nodes from T_0, and reset $T = T_0$.

The termination criterion is not on the value of Δ, but on the size of the graph: we terminate once it is reduced to a single node. The main progress is done when an abundant arc pq appears: in this case, we first set the flow value on every non-tight arc to 0, and then reduce the number of nodes by one using the above described subroutine Contract(pq). Further, the value of the scaling factor Δ is multiplied by 16, and the counter k is increased by one. The sets T_0 and T are reset to $\emptyset$. A sequence of such contractions is performed until all abundant arcs are contracted. The iterations between two phases where contractions are performed (and those up to the first contraction) will be referred to as a *major cycle* of the algorithm. In the description and the analysis, n and m will always refer to the size of the original instance and not the actual contracted one.

At termination, the subroutine Expand-to-Original finds primal and dual optimal solutions in the original graph. This is done by first expanding all contracted arcs pq by the subroutine Reverse(pq), taking these arcs in the reverse order of their contraction. Hence we obtain a dual optimal solution μ^* in the original graph (see Lemma 5.1). Finally, the subroutine Tight-flow(V, μ^*) obtains a primal optimal solution, as guaranteed by Theorem 2.6(i).

Theorem 4.3. *The algorithm* Enhanced Continuous Scaling *finds an optimal solution for the uncapacitated formulation (P) in running time $O(n^3 m^2)$ elementary arithmetic operations and comparisons.*

To get a truly strongly polynomial algorithm, we also need to guarantee that the size of the numbers during the computations remain polynomially bounded. We shall modify the algorithm in Section 6 by incorporating additional rounding steps to achieve that.

```
Algorithm Enhanced Continuous Scaling
Initialize$(V, E, b, \gamma, \bar{f})$ ;
$T_0 \leftarrow \emptyset$ ; $T \leftarrow \emptyset$ ;
$k \leftarrow 0$ ;
While $|V| > 1$ do
    $N \leftarrow \{t\} \cup \{i \in V - t : e_i^\mu < (d_i + 1)\Delta\}$ ;
    if $N \cap T \neq \emptyset$ then
        pick $p \in T_0$, $q \in N \cap T$ connected by a tight path $P$ in $E_f^\mu(\Delta)$ ;
        send $\Delta$ units of relabeled flow from $p$ to $q$ along $P$ ;
        if $e_p^\mu < (d_p + 2)\Delta$ then $T_0 \leftarrow T_0 \setminus \{p\}$ ;
        $T \leftarrow T_0$ ;
    else
        if $\exists ij \in E_f^\mu(\Delta)$, $\gamma_{ij}^\mu = 1$, $i \in T$, $j \in V \setminus T$ then $T \leftarrow T \cup \{j\}$ ;
        else
            if $\left(\forall i \in (V \setminus T) - t : |b_i^\mu| < \frac{\Delta}{16^k n}\right)$ then Filtration$(V \setminus T, f, \mu)$ ;
            if $(e_i^\mu \geq (d_i + 2)\Delta$ for all $i \in T_0)$ and $(e_i^\mu \geq (d_i + 1)\Delta$ for all $i \in T)$
                then Elementary step$(T, T_0, f, \mu, \Delta)$ ;
            elseif $\exists i \in T_0 : e_i^\mu < (d_i + 2)\Delta$ then
                $T_0 \leftarrow T_0 \setminus \{i : e_i^\mu < (d_i + 2)\Delta\}$ ;
                $T \leftarrow T_0$ ;
    while $\exists$ $pq \in E$: $f_{pq}^\mu \geq 17m\Delta$ do
        for all $ij \in E$ : $\gamma_{ij}^\mu < 1$ do $f_{ij} \leftarrow 0$ ;
        Contract$(pq)$ ;
        $\Delta \leftarrow 16\Delta$ ;
        $k \leftarrow k + 1$ ;
        $T_0 \leftarrow \emptyset$ ; $T \leftarrow \emptyset$ ;
Expand-to-Original$(\mu)$ ;
```

Figure 4: Description of the strongly polynomial algorithm

We remark that the algorithm can be simplified by terminating once the first abundant arc is found, and restarting from scratch on the contracted graph. This would give a running time bound $O(n^3 m^2 \log n)$: hence, we are able to save a factor $\log n$ by continuing with the contracted image of the current flow instead of a fresh start.

5 Analysis of the strongly polynomial algorithm

Many properties of the Continuous Scaling algorithm derived in Section 3 remain valid. In particular, Lemmas 3.1 and 3.3, and Claims 3.2 and 3.8 are applicable with repeating the proofs verbatim. The argument bounding the number of iterations will be an extension of the one in Section 3.1.

5.1 Properties of dual solutions

Let us first verify that expanding the dual optimal solution of the contracted instance results in a valid dual optimal solution of the original instance.

Lemma 5.1. *Assume that $pq \in E$ satisfies $\gamma_{pq}\mu^*_p = \mu^*_q$ for every optimal solution μ^* to (D) for the problem instance (V, E, t, b, γ). Let μ' be an optimal solution to (D) to the contracted instance $(V', E', t', b', \gamma')$ obtained by the subroutine* Contract*(pq). If $p \neq t$, then let $\mu_i := \mu'_i$ for every $i \in V - p$ and let $\mu_p := \mu'_q/\gamma_{pq}$. If $p = t$, then let $\mu_i := \mu'_i$ for every $i \in V - q$ and let $\mu_q = \gamma_{pq}$. Then μ is an optimal solution to (D) in the original instance (V, E, t, b, γ).*

Proof. We give the proof to the $p \neq t$ case only; the other case follows similarly. First, let us verify that μ is a feasible solution to (D). It is straightforward that $\mu_t = 1$ and $\mu_i > 0$ if $i \in V - t$. Also, $\gamma^\mu_{ij} \le 1$ is straightforward if $i, j \neq q$, and $\gamma^\mu_{pq} = 1$. For an arc $ip \in E$, let $iq \in E'$ denote its image. Then $\gamma'_{iq}\frac{\mu'_i}{\mu'_q} \le 1$, which can be written as $\gamma_{ip}\gamma_{pq}\frac{\mu_i}{\mu_p\gamma_{pq}} \le 1$, giving $\gamma^\mu_{ip} \le 1$. One can verify $\gamma^\mu_{pi} \le 1$ for every $pi \in E$ analogously.

Assume for a contradiction that μ is not optimal to (D): there exists an optimal solution μ^* with $\sum_{i\in V} b^{\mu^*}_i > \sum_{i\in V} b^\mu_i$. By our assumption, $\gamma_{pq}\mu^*_p = \mu^*_q$ must hold. Consider the restriction of μ^* to $V' = V \setminus \{p\}$; it is easy to check that it is feasible to (D) in the contracted instance. Using $b'_p = b_p + \gamma_{pq}b_q$, and thus $b'^{\mu^*}_s = b^{\mu^*}_p + b^{\mu^*}_q$, and $b'^{\mu'}_s = b^\mu_p + b^\mu_q$, we obtain a contradiction by

$$\sum_{i\in V} b^{\mu'}_i < \sum_{i\in V} b^{\mu^*}_i = \sum_{i\in V'} b'^{\mu^*}_i \le \sum_{i\in V'} b'^{\mu'}_i = \sum_{i\in V} b_i{}^{\mu'}.$$

□

Our next claim justifies that the feasibility properties are maintained during the algorithm.

Claim 5.2. *Let $\Delta' := 16\Delta$, and let f' and μ' denote the flow and labels after contracting the abundant arc pq. Then μ' is a conservative labeling for f', with $e^{\mu'}_i(f') < (d_i + 2)\Delta'$ for all $i \in V - t$.*

Proof. Before the contraction, the flow on every non-tight arcs is set to 0; this increases e^μ_i on every node by at most $d_i\Delta$. Let $s = p$ or $s = q$ denote the contracted node. It is straightforward by the properties of the contraction that if e' is the image of the arc e, then $\gamma^{\mu'}_{e'} = \gamma^\mu_e$. Since μ is conservative for f before the contraction, it follows that μ' is conservative for f'.

Consider a node $i \neq s$. Setting the flow values on non-tight arcs to 0 increased e^μ_i by at most $d_i\Delta$, and $e^{\mu'}_i(f') = e^\mu_i(f)$, and hence $e^{\mu'}_i(f') \le (5d_i + 8)\Delta < (d_i + 2)\Delta'$. Let us now consider the contracted node s. There is nothing to prove about $e^{\mu'}_s(f')$ if $s = t$, hence we may assume $s \neq t$. Before the contraction, we had $e^\mu_p(f) \le (5d_p + 8)\Delta$, $e^\mu_q(f) \le (5d_q + 8)\Delta$, and it is easy to verify that $e^{\mu'}_s(f') = e^\mu_p(f) + e^\mu_q(f) \le (5d_p + 5d_q + 16)\Delta$. Note that $d_s \ge \max\{d_p, d_q\} - 1 \ge \frac{d_p+d_q}{2} - 1$, implying that $e^{\mu'}_s(f') \le (d_s + 2)\Delta'$, as required. □

5.2 Bounding the number of iterations

Recall the notions of shrinking, expanding and neutral iterations from Section 3.1. We shall prove the following bound.

Theorem 5.3. *The total number of iterations in* Enhanced Continuous Scaling *is at most $390n^3m$, among them at most $195n^2m$ shrinking ones.*

The ground set V changes due to the arc contractions. Let us say that a node s is *born* in iteration $\tau+1$ if $s \in \{p,q\}$ for an abundant arc contracted in iteration τ; the original nodes are born in iteration 1. Note that we keep the same notation p or q for the new node. Further, we say that a node is *alive* until the first iteration when an incident arc gets contracted, when it *dies.* Also note that multiple contractions may happen in the same iteration; in this case, some nodes die immediately after they are born; such nodes will be ignored in the analysis. A key quantity in the analysis is

$$\Gamma_i := \log_2 \frac{32mn\Delta}{|b_i^\mu|},$$

for all nodes $i \in V-t$. Let $\Gamma_i^{(\tau)}$ denote the value at the beginning of iteration τ. We first show that $\Gamma_i \geq 0$ must hold for every $i \in V-t$, as otherwise some abundant arcs would appear.

Claim 5.4. *$\Gamma_i \geq 0$ holds for all $i \in V-t$ in every iteration after the first one.*

Proof. Assume $\Gamma_i \leq 0$, that is, $|b_i^\mu| \geq 32mn\Delta$ holds for some node $i \in V-t$ at a certain iteration after the first one. We show that there is an abundant incoming or outgoing arc incident to i. This contradicts the fact that all such arcs were contracted at the end of the previous iteration. Since f is generalized flow in every iteration, we have $e_i^\mu \geq 0$. If there are no abundant arcs incident, then $f_{ji}^\mu < 17m\Delta$ on every incoming arc ji and $f_{ij}^\mu < 17m\Delta$ on all outgoing arcs ij. First, consider the case when $b_i^\mu > 0$. Now

$$0 \leq e_i^\mu = \sum_{j:ji\in E} \gamma_{ji}^\mu f_{ji}^\mu - \sum_{j:ij\in E} f_{ij}^\mu - b_i^\mu < 17d_i m\Delta - 32nm\Delta < 0$$

a contradiction. On the other hand, if $b_i^\mu < 0$, then

$$(4d_i+8)\Delta \geq e_i^\mu = \sum_{j:ji\in E} \gamma_{ji}^\mu f_{ji}^\mu - \sum_{j:ij\in E} f_{ij}^\mu - b_i^\mu > -17d_i m\Delta + 32nm\Delta \geq (15nm+17m)\Delta,$$

using $d_i \leq n-1$. This is a contradiction since $m \geq n \geq d_i + 1$. □

Let us introduce the following set; recall that k is the number of abundant arcs contracted so far.

$$D := \left\{ i \in V-t : |b_i^\mu| \geq \frac{\Delta}{16^k n} \right\}. \tag{9}$$

Let $D^{(\tau)}$ denote this set at the beginning of iteration τ. Note that the condition for calling FILTRATION in the algorithm is precisely $(V \setminus T) \cap D = \emptyset$.

Lemma 5.5. *(i) The $\Gamma_i^{(\tau)}$ values are monotone decreasing inside every major cycle, and they increase by 4 when an abundant arc is contracted.*

(ii) After the contraction of k abundant arcs,

$$\Gamma_i^{(\tau)} \leq 4k + 5 + 4\log_2 n$$

holds for every $i \in D^{(\tau)}$.

(iii) $D^{(\tau)} \subseteq D^{(\tau+1)}$ inside a major cycle. When an abundant arc pq is contracted at the end of iteration τ, then $D^{(\tau)} \setminus \{p,q\} \subseteq D^{(\tau+1)} \setminus \{p,q\}$.

Proof. Inside a major cycle of the algorithm, the ratio $|b_i^\mu|/\Delta$ can never decrease: in ELEMENTARY STEP(T, T_0, f, μ, Δ), it is unchanged for $i \in T$ and increases for $i \in V \setminus T$. At the end of a major cycle, every ratio $|b_i^\mu|/\Delta$ decreases by a factor of 16. This proves *(i)*. Part *(ii)* is straightforward by $|b_i^\mu| \geq \Delta/(16^k n)$ and $\log_2(mn^2) \leq 4\log_2 n$.

For part *(iii)*, it is straightforward that if no arcs are contracted, then no node may leave D. Further, when an abundant arc is contracted, the threshold in the definition of D is unchanged since $\Delta/16^k = (16\Delta)/16^{k+1}$. Therefore if $i \in D \setminus \{p, q\}$ before the contraction, then i remains in D after the contraction. □

Let us introduce some further classification of iterations. Let $\mathcal{C}$ denote the set of iterations when contractions are performed. Clearly, $|\mathcal{C}| \leq n - 1$. Let $\mathcal{F}$ denote the set of iterations when the subroutine FILTRATION is performed; such iterations will be called *filtrating*. Notice that $\tau \in \mathcal{F}$, that is, iteration τ is filtrating if and only if $(V \setminus T^{(\tau)}) \cap D^{(\tau)} = \emptyset$. Let $\mathcal{D}$ denote the set of iterations τ when D is extended: $D^{(\tau)} \subsetneq D^{(\tau+1)}$. By the above claim, this may happen at most $2n - 1$ times, as every node may enter D only once during its lifetime. Hence $|\mathcal{D}| \leq 2n - 1$. Let us define

$$\Gamma^{(\tau)} := \sum_{i \in D^{(\tau)}} \Gamma_i^{(\tau)}$$

Claim 5.6. *During the entire algorithm, the total increase in the value of* $\Gamma^{(\tau)}$ *can be bounded by* $14n^2$.

Proof. When a node i enters D after the contraction of k arcs, by Lemma 5.5(ii) we have $\Gamma_i \leq 4k + 5 + 4\log_2 n$. There are $\leq n - 1 - k$ more contractions, accounting for a total increase of $\leq 4(n - 1 - k)$ in all later iterations. Hence the total increase for a node i is bounded by $4n + 1 + 4\log_2 n \leq 7n$. On the other hand, there are altogether $\leq 2n - 1$ nodes born during the entire algorithm. □

The following claim is straightforward, since for every $i \in V \setminus T$, b_i^μ is unchanged during ELEMENTARY STEP(T, T_0, f, μ, Δ), whereas Δ decreases by a factor α.

Claim 5.7. *If iteration* $\tau \notin \mathcal{F}$, *then for at least one* $i \in D^{(\tau)}$, *the* Γ_i *value decreases by* $\log_2 \alpha^{(\tau)}$.

Together with Claim 5.6, it yields the following.

Lemma 5.8. *During the entire algorithm, we have*

$$\sum_{\tau \notin \mathcal{C} \cup \mathcal{F}} \log_2 \alpha^{(\tau)} \leq 14n^2,$$

Proof. The right hand side bounds the total increase in Γ according to Claim 5.6. By the previous claim, at least one $\Gamma_i^{(\tau)}$ decreases by at least $\log_2 \alpha^{(\tau)}$ in iteration $\tau \notin \mathcal{F}$. By Claim 5.4, $\Gamma_i^{(\tau+1)} \geq 0$ and therefore $\Gamma_i^{(\tau)} \geq \log_2 \alpha^{(\tau)}$, as otherwise an abundant arc incident to i should have been contracted at the end of iteration τ, giving $\tau \in \mathcal{C}$. □

The following lemma is the analogue of Lemma 3.7.

Lemma 5.9. *While alive, every node* $i \in V - t$ *may enter the set* T_0 *at most* $|\mathcal{D}| + \sum_{\tau \notin \mathcal{C} \cup \mathcal{F}} \log_2 \alpha^{(\tau)}$ *times.*

Before proving the lemma, let us show how it can be used to bound the total number of iterations.

Proof of Theorem 5.3. The proof follows the same lines as that of Lemma 3.6, analyzing the invariant Ψ as defined by (7). Consider an iteration $\tau \in \mathcal{C}$ when some abundant arcs are contracted. According to Claim 5.2, the value of Ψ decreases to 0 in all such iterations.

Every shrinking iteration decreases Ψ by one, and the only steps when Ψ increases is when some node $i \in V - t$ enters T_0. Let λ_i denote the number of times this happens. Lemmas 5.8 and 5.9 imply $\lambda_i \le |\mathcal{D}| + 14n^2 \le 2n + 14n^2 \le 15n^2$. Consequently, the total increase in Ψ is bounded by

$$\sum_{i \in V-t} (3d_i + 7)\lambda_i \le 15n^2 \sum_{i \in V-t} (3d_i + 7) \le 15n^2(6m + 7n) \le 195n^2 m.$$

As in the proof of Lemma 3.6, this bounds the number of shrinking iterations, and there can be $\le 2n$ iterations between two subsequent shrinking iterations. This completes the proof. □

The next claims are needed for the proof of Lemma 5.9.

Claim 5.10. *Consider a filtrating iteration $\tau \in \mathcal{F}$. The maximum flow problem in* Filtration$(V \setminus T, f, \mu)$ *is feasible, and after the subroutine, every $i \in V \setminus T$ satisfies*

$$e_i^\mu \le R_i^\mu + n \max_{j \in (V \setminus T) - t} |b_j^\mu|.$$

Proof. Feasibility is verified by the restriction of $f^{(\tau)}$ to tight arcs in $E[V \setminus T]$. This gives a feasible solution as in the proof of Lemma 3.9; note that the arcs entering $V \setminus T$ are all non-tight, as otherwise we would have extended T in this iteration instead. Let f' denote the generalized flow on $V \setminus T$ returned by Tight Flow$(V \setminus T, \mu)$, and f the generalized flow returned by Filtration$(V \setminus T, f, \mu)$. Inside $E[V \setminus T]$, f is nonzero only on tight arcs, and equals $f_{ij} = f'_{ij}$ for all $ij \in E[V \setminus T]$. The value of f is set to zero on arcs leaving $V \setminus T$ and the original values $f_{ij}^{(\tau)}$ are kept if $i \in T$. We obtain $e_i^\mu(f) = e_i^\mu(f') + R_i^\mu$ for $i \in V \setminus T$, since the non-tight arcs are precisely those coming from T. The claim then follows by Theorem 2.6(ii). □

Lemma 5.11. *Let $\tau \in \mathcal{F} \setminus \mathcal{C}$ be a filtrating iteration when no contraction is performed.*

(i) If $\beta_i^{(\tau+1)} \ge (d_i + 1)$ for some $i \in V \setminus T^{(\tau)}$, then $\tau \in \mathcal{D}$, that is, $D^{(\tau+1)} \supsetneq D^{(\tau)}$.

(ii) Either $\tau \in \mathcal{D}$, or one of the iterations τ, $(\tau + 1)$ and $(\tau + 2)$ must be shrinking.

Proof. (i): Let $\Delta = \Delta^{(\tau)}$ and $T = T^{(\tau)}$. First, let us prove that Elementary step(T, T_0, f, μ, Δ) must have been performed in iteration τ. This follows by Claim 5.10. Indeed, if Elementary step(T, T_0, f, μ, Δ) is skipped after calling Filtration$(V \setminus T, f, \mu)$, then for every $i \in V \setminus T$ we have

$$e_i^\mu \le R_i^\mu + n \max_{j \in (V \setminus T) - t} |b_j^\mu| < (d_i + 1)\Delta$$

at the beginning of iteration τ+1. This follows by Claim 2.5 ($R_i^\mu < d_i \Delta$), and since $\max_{j \in (V \setminus T) - t} |b_j^\mu| < \Delta / n$ by $(V \setminus T) \cap D^{(\tau)} = \emptyset$. This is a contradiction to $\beta_i^{(\tau+1)} \ge (d_i + 1)$. This shows Elementary step(T, T_0, f, μ, Δ) must have been performed in iteration τ, setting $\Delta' = \Delta^{(\tau+1)} = \Delta^{(\tau)} / \alpha^{(\tau)}$ (note that we assumed $\tau \notin \mathcal{C}$ as well).

Consider a node $i \in V \setminus T$ in iteration τ for which β_i increased above $(d_i + 1)$. After Filtration$(V \setminus T, f, \mu)$, $F^\mu[V \setminus T] = \emptyset$ and $f_{pq} = 0$ for every $pq \in E[V \setminus T, T]$. Therefore Elementary step(T, T_0, f, μ, Δ) does not change the flow f at all; also by definition, the labels μ_i are unchanged for $i \in V \setminus T$. Hence e_i^μ and b_i^μ do not change for $i \in V \setminus T$. Let Δ and Δ' denote the scaling factor before and after ELEMENTARY STEP(T, T_0, f, μ, Δ). We have

$$d_i + 1 \le \frac{e_i^\mu}{\Delta'} \le \frac{R_i^\mu + n \max_{j \in (V \setminus T) - t} |b_j^\mu|}{\Delta'} \le d_i + \frac{n \max_{j \in (V \setminus T) - t} |b_j^\mu|}{\Delta'}.$$

In the second inequality we use that R_i^μ is unchanged in ELEMENTARY STEP(T, T_0, f, μ, Δ) and it must be at most $d_i \Delta'$ by Claim 2.5. This implies $\Delta'/n \le \max_{j \in (V \setminus T) - t} |b_j^\mu|$. Since $(V \setminus T) \cap D^{(\tau)} = \emptyset$ was assumed, it follows that D must be extended in this iteration, that is, $\tau \in \mathcal{D}$.

For part *(ii)*, assume $\tau \notin \mathcal{D}$. Some nodes $i \in T_0$ might be removed in iteration τ if e_i^μ decreases below $(d_i + 2)$; in this case, iteration τ itself is shrinking. Otherwise, part *(i)* implies that $V \setminus T^{(\tau)} \subseteq N^{(\tau+1)}$, and that $\alpha = \alpha_2$ in iteration τ. Therefore at the beginning of iteration $\tau + 1$, there exists a tight arc $ij \in E$ with $i \in T^{(t)}$, $j \in V \setminus T^{(t)}$. Now either $T^{(\tau)} \cap N^{(\tau)} \neq \emptyset$ already holds, in which case a path augmentation is performed; or iteration $\tau + 1$ extends T using the tight arc ij. In this case, $j \in T^{(\tau+2)} \cap N^{(\tau+2)}$, and iteration $\tau + 2$ is shrinking. □

We are ready to prove Lemma 5.9. The proof is based on that of Lemma 3.7, also making use of the above claims.

Proof of Lemma 5.9. Let $\tau_1 < \tau_2 < \ldots < \tau_\lambda$ denote the iterations when i enters T_0. This number is not necessarily finite; hence $\lambda = \infty$ is allowed. We have $\beta_i^{(\tau_\ell + 1)} = 4(d_i + 2)$ for $1 \le \ell \le \lambda$.

For $1 \le \ell \le \lambda$, let us define τ'_ℓ to be the largest value $\tau'_\ell \le \tau_\ell$ such that $\beta_i^{(\tau'_\ell)} < (d_i + 2)$. The existence of these values follows as in the proof of Lemma 3.7.

In iteration τ'_ℓ, we have $i \notin T_0$, since once the excess e_i drops below $(d_i + 2)\Delta\mu_i$, the node i is immediately removed from T_0. Also, i will be added to T_0 in iteration τ_ℓ. We claim that $\mathcal{C} \cap [\tau'_\ell, \tau_\ell] = \emptyset$. Indeed, if $\theta \in \mathcal{C}$, then $\beta_i^{(\theta+1)} < (d_i + 2)$ by Claim 5.2; this would contradict the maximal choice of τ'_ℓ.

Let us analyze the case when $\mathcal{D} \cap [\tau'_\ell, \tau_\ell] = \emptyset$ holds. According to Lemma 5.11(i), this implies $\mathcal{F} \cap [\tau'_\ell, \tau_\ell] = \emptyset$. Indeed, if $\theta \in \mathcal{F} \cap [\tau'_\ell, \tau_\ell]$, then $\beta_i^{(\theta+1)} < (d_i + 1)$ would follow, a contradiction again to the maximal choice of τ'_ℓ.

With the same argument as in the proof of Lemma 3.7, making use of Claim 3.8, we obtain

$$4 \le \left(\prod_{\theta \in [\tau'_\ell, \tau_\ell]} \alpha^{(\theta)} \right)^2 .$$

Note that we have $[\tau'_\ell, \tau_\ell] \cap (\mathcal{C} \cup \mathcal{F} \cup \mathcal{D}) = \emptyset$. Let us add the logarithms of these inequalities for those values $\ell = 1, \ldots, \lambda$ where $[\tau'_\ell, \tau_\ell] \cap \mathcal{D} = \emptyset$. Hence we obtain

$$\lambda - |\mathcal{D}| \le \sum_{\theta \notin \mathcal{C} \cup \mathcal{F}} \log_2 \alpha^{(\theta)}$$

completing the proof. □

5.3 Running time analysis

Proof of Theorem 4.3. As shown in Theorem 5.3, the total number of shrinking steps is $O(n^2 m)$. If FILTRATION is not called between two shrinking iterations, then this part of the algorithm can be implemented in $O(m + n \log n)$ time using Fibonacci heaps, using the variant described in Section 6. If FILTRATION is called, then we must execute a maximum flow computation in $O(nm)$ time [24].

According to Lemma 5.11(ii), in this case we must have a shrinking one within the next three iterations. Consequently, the running time between two shrinking iterations is dominated by $O(nm)$. This gives a total estimation of $O(n^3m^2)$; all other steps of the algorithm (contractions, initial and final flow computations, etc.) are dominated by this term. □

6 Bounding the encoding size

In this section, we complete the proof of Theorem 2.1: we modify the algorithm to guarantee that the encoding size of the numbers during the computations remain polynomially bounded in the input size. Further, we present a more efficient implementation, by jointly performing the elementary steps between two shrinking iterations; this enables a better running time bound, as already indicated in the proofs of Theorems 2.7 and 4.3. We describe the modifications for the Enhanced Continuous Scaling algorithm, but they are naturally applicable for the weakly polynomial Continuous Scaling algorithm as well.

In this section, let us assume that

$$\bar{B} \geq 500n^5. \tag{10}$$

Indeed, if $\bar{B}$ is polynomially bounded in n, then any of the previous weakly polynomial algorithms has strongly polynomial running time. We define the following quantities needed for the roundings; as in the previous section, n and m will always refer to the size of the original input instance (and not the actual contracted one).

$$q := 40m\bar{B}^4, \quad \bar{q} := 40m\bar{B}^2 = q/\bar{B}^2$$

For a real number $a \in \mathbb{R}_{\geq 0}$, let $\lfloor a \rfloor_q$ denote the largest number p/q with $p \in \mathbb{Z}$, $p/q \leq a$, and similarly, let $\lceil a \rceil_q$ denote the smallest number p/q with $p \in \mathbb{Z}$, $p/q \geq a$. The same notation will also be used for $\bar{q}$.

The main subroutine Aggregate steps(T_0, f, μ, Δ) is shown in Figure 5. The input is a set T_0 with $e_i^\mu \geq (d_i + 2)\Delta$ for every $i \in T_0$. The output is either a node $s \in V$ or $s = NULL$. If $s \neq NULL$, then we can find a tight path in $E_f^\mu(\Delta)$ between a node $p \in T_0$ and s, where either $s = t$ or $e_s^\mu < (d_s + 1)\Delta$. The case $s = NULL$ means that some node i leaves T_0, that is, e_i^μ drops below $(d_i + 2)\Delta$. In the first case, we can perform a path augmentation from s to t. The entire algorithm is exhibited in Figure 6. Note that termination can happen either because the graph is shrunk to a single node, or because Δ decreases below a certain threshold as in the weakly polynomial algorithm Continuous Scaling.

We now describe the main features of the subroutine Aggregate steps and compare it to the algorithm Enhanced Continuous Scaling. Apart from the rounding and contraction steps, it performs essentially the same as a sequence of Elementary steps starting with $T = T_0$, until a next shrinking iteration. The difference is that in Enhanced Continuous Scaling, whenever the set T is extended by a node, Elementary step needs to update the labels of every node in T and change flow values on certain arcs. During a sequence of iterations between two shrinking steps, this can lead to $O(n)$ value updates in certain nodes and arcs. In contrast, Aggregate steps changes the labels only once and the flow values at most twice. The key quantity in Aggregate steps is α^*. This corresponds to the product of the α multipliers of the sequence of Elementary steps thus far in Enhanced Continuous Scaling.

The subroutine Aggregate steps starts with $T = T_0$ and extends T by adding nodes one-by-one, until $t \in T$, or $e_s^\mu < (d_s + 1)\Delta$ for some $s \in T \setminus T_0$, or $e_i^\mu < (d_i + 2)\Delta$ for some $i \in T_0$. Node labels are changed at the end of the subroutine only. For a node $i \in T$, we store the value α_i at the

```
Subroutine Aggregate steps($T_0, f, \mu, \Delta$)
for all $i \in (V \setminus T_0) - t$ do
    update $r_1(i), r_2(i), r_3(i), r_4(i)$ and $\delta_i$ as in (4) and (5) ;
    $\alpha_i \leftarrow \min\left\{\lfloor \delta_i \rfloor_q, \min\{1/\gamma^\mu_{ji} : ji \in E^\mu_f(\Delta), j \in T_0\}\right\}$ ;
$\alpha_t \leftarrow \min\{1/\gamma^\mu_{jt} : jt \in E^\mu_f(\Delta), j \in T_0\}$ ;
for all $i \in T_0$ do
    update $\rho_i, \nu_i$ as in (11) ; $\alpha_i \leftarrow 1$ ;
$T \leftarrow T_0$ ; $\alpha^* \leftarrow 1$ ; $\lambda \leftarrow \min\{\nu_j : j \in T_0\}$ ;
while $(\alpha^* \leq \lambda)$ and $(t \notin T)$ do
    $\alpha^* \leftarrow \min\{\alpha_i : i \in V \setminus T\}$ ;
    if $\alpha^* = \infty$ then
        set $f_{ti} = 0$ for all $ti \in E$: $\gamma^\mu_{ti} < 1$ ;
        return optimal flow $f$ and optimal relabeling $\mu$ ; TERMINATE.
    $i \leftarrow \mathrm{argmin}\{\alpha_i : i \in V \setminus T\}$ ;
    $T \leftarrow T \cup \{i\}$ ;
    if $\alpha_i = \lfloor \delta_i \rfloor_q$ then $T_0 \leftarrow T_0 \cup \{i\}$ ;
    for all $ij \in E : j \in T$ do
        $f_{ij} \leftarrow f_{ij}/\alpha^*$ ;
        update $\rho_j, \nu_j$ as in (11) ;
        $\lambda \leftarrow \min\{\lambda, \nu_j\}$ ;
    for all $ij \in E : j \in V \setminus T$ do
        if $\gamma^\mu_{ij} < 1$ then $f_{ij} \leftarrow f_{ij}/\alpha^*$ ;
        if $j \neq t$ then
            update $r_1(j), r_2(j), \delta_j$ as in (4) and (5) ;
            $\alpha_j \leftarrow \min\left\{\alpha_j, \lfloor \delta_j \rfloor_q, \alpha_i/\gamma^\mu_{ij}\right\}$ ;
        if $j = t$ then $\alpha_t \leftarrow \min\{\alpha_t, \alpha_i/\gamma^\mu_{it}\}$ ;
    for all $ji \in E : j \in V \setminus T$ do
        if $\gamma^\mu_{ji} = 1$ then
            $f_{ji} \leftarrow f_{ji}\alpha^*$ ;
            if $f_{ji} > \Delta\mu_j$ then $\alpha_j \leftarrow \alpha^*$ ;
        if $j \neq t$ then
            update $r_3(j), r_4(j), \delta_j$ as in (4) and (5) ;
            $\alpha_j \leftarrow \min\{\alpha_j, \lfloor \delta_j \rfloor_q\}$ ;
    update $\rho_i, \nu_i$ as in (11) ;
    $\lambda \leftarrow \min\{\lambda, \nu_i\}$ ;
    if $\left(\forall j \in (V \setminus T) - t : |b^\mu_j| < \frac{\Delta}{16^k n(1+1/\bar{B})^g}\right)$ then
        Filtration($V \setminus T, f, \mu$) ;
        for all $j \in T$ do
            $e_j \leftarrow e_j - \rho_j$ ; $\rho_j \leftarrow 0$ ;
            update $\nu_j$ as in (11) ;
        for all $j \in V \setminus T$ do
            update $r_1(i), r_2(i), r_3(i), r_4(i), \delta_i$ as in (4) and (5) ;
            $\alpha_i \leftarrow \min\{\lfloor \delta_i \rfloor_q, \alpha_i\}$ ;
        $\lambda \leftarrow \min\{\nu_j : j \in T\}$ ;
$\Delta \leftarrow \lceil \Delta/\alpha^* \rceil_q$ ;
for all $ij \in F^\mu[V \setminus T] \cup E[V \setminus T, T]$ do $f_{ij} \leftarrow f_{ij}/\alpha^*$ ;
for all $i \in T$ do $\mu_i \leftarrow \mu_i\alpha^*/\alpha_i$ ;
for all $j \in T_0 : \nu_j < \alpha^*$ do $T_0 \leftarrow T_0 \setminus \{j\}$ ;
Round Label($f, \mu$) ;
if $t \in T$ then RETURN $s = t$ ; elseif $\exists s \in T \setminus T_0 : \nu_s < \alpha^*$ then RETURN such an $s$ ;
otherwise RETURN $s = NULL$.
```

```
Algorithm Modified Enhanced Continuous Scaling
Initialize;
$T_0 \leftarrow \emptyset$ ; $k \leftarrow 0$ ; $g \leftarrow 0$ ;
While $|V| > 1$ and $\Delta \geq 1/(17m\bar{B}^3)$ do
    $s \leftarrow$ Aggregate Steps$(T_0, f, \mu, \Delta)$ ;
    $g \leftarrow g + 1$ ;
    if $s \neq NULL$ then
        pick a tight $p - s$ path $P$ in $E_f^\mu(\Delta)$ with $p \in T_0$ ;
        send $\Delta$ units of relabeled flow from $p$ to $s$ along $P$;
        if $e_p^\mu < (d_p + 2)\Delta$ then $T_0 \leftarrow T_0 \setminus \{p\}$ ;
    while $\exists$ $pq \in E$: $f_{pq}^\mu \geq 17m\Delta$ do
        for all $ij \in E$ : $\gamma_{ij}^\mu < 1$ do $f_{ij} \leftarrow 0$ ;
        Contract$(pq)$ ;
        $\Delta \leftarrow 16\Delta$ ;
        $k \leftarrow k + 1$ ;
        $T_0 \leftarrow \emptyset$ ;
if $\Delta < 1/(17m\bar{B}^3)$ then Tight-Flow$(V, \mu)$
    else Expand-to-Original$(\mu)$ ;
```

Figure 6: Description of the modified strongly polynomial algorithm

time when i enters T, and multiply the label μ_i at the end by α^*/α_i. The flow on an arc ij may change when its endpoints enter T, or at the end of the subroutine, altogether at most twice.

For nodes $i \in V \setminus T$, we use α_i to denote the candidate value of α^* when i must enter T, either due to a new tight arc $ji \in E_f^\mu(\Delta)$, $j \in T$, or because the excess e_i reaches the threshold $4(d_i+2)\Delta\mu_i$ and hence i must be included in T_0. We define δ_i as in (5), representing the value of α^* when i would enter T because of $e_i = 4(d_i + 2)\Delta\mu_i$, provided that no other node enters T before. We let

$$\alpha_i := \min\left\{\lfloor\delta_i\rfloor_q, \min\{\alpha_j/\gamma_{ji}^\mu : ij \in E_f^\mu(\Delta), j \in T\}\right\}.$$

Note the rounding $\lfloor\delta_i\rfloor_q$ in the first case. This means that e_i might be slightly less than $4(d_i+2)\Delta\mu_i$ when i enters T. The second event corresponds to the case when i enters T due to a new tight arc from a node $j \in T$. Note that either $ji \in E$, or ji is a reverse arc with $ij \in E$, $\gamma_{ij}^\mu = 1$, $f_{ij}^\mu > \Delta$. In the latter case the corresponding term equals α_j.

For $i \in T \setminus T_0$, wish to estimate when $e_i < (d_i + 1)\Delta\mu_i$ would be attained, and for $i \in T_0$, we wish to estimate when $e_i < (d_i + 2)\Delta\mu_i$ would be attained. To provide a unified notation for these two cases, let us define $\xi_i = 1$ if $i \in T \setminus T_0$ and $\xi_i = 2$ if $i \in T_0$. We let

$$\begin{aligned} \rho_i &:= \sum_{j \in V \setminus T} \gamma_{ji} f_{ji}, \\ \nu_i &:= \begin{cases} \infty & \text{if } e_i - \rho_i \geq (d_i + \xi_i)\Delta\mu_i; \\ \frac{\rho_i}{(d_i+\xi_i)\Delta\mu_i + \rho_i - e_i} & \text{otherwise.} \end{cases} \end{aligned} \tag{11}$$

Here ρ_i denotes the total flow entering i on arcs from $V \setminus T$, these are the ones where the flow value will be reduced. We define ν_i as the smallest value of α^* when $e_i = (d_i + \xi_i)\Delta\mu_i$ is reached. λ will denote the minimum value of $\{\nu_i : i \in T\}$. The iterations terminate once $\lambda < \alpha^*$.

In every iteration, we set the new value $\alpha^* := \min\{\alpha_i : i \in V \setminus T\}$, pick a node i minimizing this value, and include it into T. The modifications of the incident f_{ij} and f_{ji} values are in order to guarantee the same change as in the sequence of Elementary step operations in Enhanced Continous Scaling. We update the corresponding $r_1(j), \dots, r_4(j), \delta_j$ and e_j, ρ_j, ν_j values on the neighbours of i accordingly. These updates can be performed in $O(1)$ time. Indeed, for each of the sums $r_1(j), \dots, r_4(j), e_j, \rho_j$, only one term changes. Provided these, δ_j and ν_j are obtained by simple formulae.

The Filtration subroutine is used similarly as in Enhanced Continuous Scaling, but the bound on $|b^\mu_j|$ is different, containing a term $(1 + 1/\bar{B})^g$ necessary due to the roundings. The counter g denotes the total number of times the subroutine Aggregate steps was performed.

Compared to Enhanced Continuous Scaling, there is a further minor difference regarding contractions. In Enhanced Continuous Scaling, a contraction can be performed after every Elementary step, whereas in the modified algorithm, only after the entire sequence represented by Aggregate steps(T_0, f, μ, Δ).

If Filtration is not called, then the subroutine Aggregate Steps can be implemented in $O(m + n \log n)$ time using the Fibonacci heap data structure. To see this, we maintain two heap structures, one for the α_i's for $i \in V \setminus T$, an one for the ν_i's, $i \in T$. Besides, we maintain the $r_1(i), \dots, r_4(i)$ values for $i \in V \setminus T$, and the e_i, ρ_i values for $i \in T$. Every arc is examined $O(1)$ times, and the corresponding key modifications can be implemented in $O(1)$ time. Consequently, the bound in [7] is applicable.

It is easy to verify that all μ_i and f_{ij} values are modified exactly as in a sequence of Elementary step operations. For example, consider an arc ji with originally $i, j \in V \setminus T$, such that i enters T before j. The scaling factor when i enters T is Δ/α_i. If $ij \in E^\mu_f(\Delta/\alpha_i)$, that is, $f^\mu_{ji} > \Delta/\alpha_i$, then j enters T in the next neutral phase. Accordingly, Aggregate Steps sets $\alpha_j = \alpha^*$ in the same case. If ji was a non-tight arc already at the beginning, then f_{ji} is decreased in every elementary step until j enters T; in our subroutine, f_{ji} is divided by α_j. However, if ji was tight initially, and $f^\mu_{ji} < \Delta\alpha_i$, then it becomes non-tight after i enters T. Notice that in this case our subroutine divides f_{ji} by α_j/α_i. The other cases can be verified similarly.

At termination, we perform the subroutine Round Label, shown in Figure 7. This is a Dijkstra-type algorithm that takes labeling μ, and changes it to a labeling $\mu' \geq \mu$ such that the set of tight arcs in E_f may only increase. Consequently, if (f, μ) is a Δ-feasible pair for some Δ, then so is (f, μ').

We repeatedly extend the set S starting from $S = \{t\}$ until $S = V$ is achieved. In every iteration we multiply all μ_i's for $i \in V \setminus S$ by $\varepsilon > 1$, so that either a new tight arc between $V \setminus S$ and S is created, or some value μ_i for $i \in V \setminus S$ becomes an integer multiple of $1/\bar{q}$.

6.1 Analysis

It is easy to adapt Theorem 2.6 and Lemma 3.9 to show that if in any contracted graph during the algorithm Modified Enhanced Continuous Scaling, we have $\Delta \leq 1/(17m\bar{B}^3)$ for the original values of B, m and n, then the current labeling μ is optimal and thus we may terminate. Also note that $2\bar{B}/q \leq 1/(17m\bar{B}^3)$, and therefore we may assume that $2\bar{B}/q \leq \Delta$ in all iterations of the algorithm except the last one.

Claim 6.1. *The subroutine* Round Label *returns a labeling μ' such that every μ'_i is an integer multiple of $\bar{B}/q$. If (f, μ) is Δ-conservative for some $\Delta \geq 0$, then so is (f, μ'). Finally, $\mu_i \leq \mu'_i \leq \big(1 + 1/(40m\bar{B})\big)\,\mu_i$.*

```
Subroutine ROUND LABEL($f, \mu$)
$S \leftarrow \{t\}$ ;
while $S \neq V$ do
    $\varepsilon_1 \leftarrow \min\left\{\frac{\lceil \mu_i \rceil_{\bar{q}}}{\mu_i} : i \in V \setminus S\right\}$ ;
    $\varepsilon_2 \leftarrow \min\left\{\frac{1}{\gamma^{\mu}_{ij}} : ij \in E_f, i \in V \setminus S, j \in S\right\}$ ;
    $\varepsilon \leftarrow \min\{\varepsilon_1, \varepsilon_2\}$ ;
    for $i \in V \setminus S$ do $\mu \leftarrow \mu/\varepsilon$ ;
    $S \leftarrow S \cup \{i \in V \setminus S : \lceil \mu_i \rceil_{\bar{q}} = \mu_i\} \cup \{i \in V \setminus S : \exists j \in S, ij \in E_f, \gamma^{\mu}_{ij} = 1\}$ ;
```

Figure 7: The Round Label subroutine

Proof. A node i enters S either if μ_i is an integer multiple of $1/\bar{q} = \bar{B}^2/q$, or if it is connected by a tight path P in E_f to a node j such that μ_j is an integer multiple of $1/\bar{q}$. In the latter case, $\mu_i = \mu_j/\gamma(P)$, and since $\bar{B}$ is an integer multiple of $\gamma(P)$ by definition of $\bar{B}$ in Section 2.1, it follows that μ_i is an integer multiple of $\bar{B}/q$. The claim on conservativeness follows since every tight arc in E_f remains tight. Finally, it is clear that $\mu'_i \leq \lceil \mu_i \rceil_{\bar{q}} < \mu_i + 1/\bar{q} = \mu_i(1 + 1/(\bar{q}\mu_i))$. On the other hand, $\mu_i \geq 1/\bar{B}$ because of the initial definition (3), and hence $1 + 1/(\bar{q}\mu_i) \leq \big(1 + 1/(40m\bar{B})\big)$. □

In the original algorithm, $\Delta\mu_i$ is nonincreasing during every ELEMENTARY STEP iteration. Due to the roundings, this is not true anymore; however, we have the following bound (the possible increase corresponds to the case $\alpha_i \leq 1 + 1/\bar{B}$).

Claim 6.2. *When performing* AGGREGATE STEPS, *$\Delta\mu_i$ decreases by at least a factor of $\alpha_i/(1+1/\bar{B})$ for every $i \in T$, and by $\alpha^*/(1 + 1/\bar{B})$ for every $i \in V \setminus T$, except for possibly the ultimate iteration.*

Proof. Without the rounding, we would set the new value of the scaling factor to Δ/α^* and the new value of μ_i as $\mu_i\alpha^*/\alpha_i$ if $i \in T$ and leave it unchanged if $i \in V \setminus T$. Let us focus on the case $i \in T$; the same argument works for $i \in V \setminus T$ as well. These will be rounded to $\Delta' = \lceil \Delta/\alpha^* \rceil_q$ and $\mu'_i \leq (1 + 1/(40m\bar{B}))\mu_i\alpha^*/\alpha_i$ by the previous claim. As remarked above, we have $2\bar{B}/q \leq \Delta'$ in all save the last step of the algorithm. Therefore $\Delta' = \lceil \Delta/\alpha^* \rceil_q \leq (1 + 1/(2\bar{B}))\Delta/\alpha^*$. Consequently,

$$\Delta'\mu'_i \leq \big(1 + 1/(2\bar{B})\big)\left(1 + 1/(40m\bar{B})\right)\Delta\mu_i/\alpha_i \leq \left(1 + 1/\bar{B}\right)\Delta\mu_i/\alpha_i,$$

proving the claim. □

Provided this, one can derive the bound $O(n^2m)$ on the total number of calls to AGGREGATE STEPS as in Theorem 5.3. This subroutine corresponds to a sequence of ELEMENTARY STEP, however, the argument can be easily adapted. We now outline the changes in the analysis. Instead of (9), we define the set D as

$$D := \left\{i \in V - t : |b^{\mu}_i| \geq \frac{\Delta}{16^k n(1 + 1/\bar{B})^g}\right\}.$$

According to the above claim, if no arc is contracted, then no node may leave the set D, as in Lemma 5.5. After the contraction of k arcs, the maximum value of Γ_i can be at most

$$\Gamma^{(\tau)}_i \leq 4k + 5 + 4\log_2 n + g\log\left(1 + 1/\bar{B}\right) \leq 4k + 5 + 4\log_2 n + g/\bar{B}.$$

By the assumption (10), the last term is at most $1/n$ even after $500n^2m$ iterations. Hence the proof of Claim 5.6 can be easily modified to prove the following.

Claim 6.3. *After at most* $500n^2m$ *executions of* Aggregate steps*, the total increase in the value of* $\Gamma^{(\tau)}$ *can be bounded by* $14n^2$*.*

Another change in the argument is due to the fact that when a node i enters T_0 in Aggregate steps, it might have $e_i < 4(d_i + 2)\mu_i$ due to the rounding of δ_i. This affects the way Claim 3.8 is applied in the proof of Lemmas 3.7 and 5.9. In (8), 4 has to be replaced by a slightly smaller number; consequently, we have to replace $\log_2$ by $\log_{2-\varepsilon}$ in the argument for some small ε. However, this increases the running time estimation only by a small constant factor.

One can show that the $O(n^3m^2)$ bound on the number of elementary arithmetic operations and comparisions is still applicable for the modified algorithm. The proof of Theorem 2.1 is complete by showing that the size of the variables remain polynomially bounded. Due to the rounding steps, Δ and the μ_i's are always of polynomially bounded size. It is left to show that the same holds for the f_{ij} values.

Lemma 6.4. *Every* f_{ij} *value is a rational number of polynomially bounded size in* $\bar{B}$*.*

Proof. The f_{ij} values can be changed in two ways. One is via maximum flow computations in the initial Tight-flow subroutine and during the later Filtration iterations. We can always assume that the flow computations return a basic optimal solution; since the flow problem is defined by polynomially bounded capacities and demands, such steps reset a polynomially bounded rational value for f_{ij}.

Every Aggregate Steps iteration either leaves f_{ij} unchanged, or modifies it to f_{ij}/α_i, or to $f_{ij}\alpha_j/\alpha_i$. We claim that α_i and α_j are both integer multiples of $1/q$. Indeed, either $\alpha_i = \lfloor \delta_i \rfloor_q$ and thus this property is straightforward; or $\alpha_i = \mu_p/\gamma(P)$ for some $p - i$ path P with $p \in T_0$; note that μ_i is an integer multiple of $\bar{B}/q$ by Claim 6.1, and $\bar{B}$ is an integer multiple of $\gamma(P)$. Further, it is easy to verify that $\alpha_i, \alpha_j \le \bar{B}^2$. Consequently, f_{ij} is multiplied in Aggregate steps(T_0, f, μ, Δ) by a number Q that is the quotient of two integers $\le q\bar{B}^2$.

During a path augmentation, f_{ij} is modified by adding or subtracting $\Delta\mu_i$, that is an integer multiple of $\bar{B}/q^2$. Since Aggregate Steps is executed $O(n^2m)$ times, these arguments show that all f_{ij}'s remain polynomially bounded. □

7 Problem transformations

7.1 Transformation to an uncapacitated instance

Consider an instance $(V', E', t', u', \gamma')$ of the standard formulation (P_u) with $|V'| = n'$, $|E'| = m'$, and encoding parameter B. We now show how it can be transformed to an equivalent instance (V, E, t, b, γ) of the uncapacitated formulation (P) with $|V| \le n' + m'$, $|E| \le 2m'$, and $\bar{B} \le 2B^{4m'}$ satisfying assumptions $(\star)$, $(\star\star)$, $(\star\star\star)$, and all assumptions on the encoding size in Section 2.1. The transformation proceeds in three steps. First, we remove all arc capacities by introducing new nodes for arcs with finite capacities. In the second step, the boundedness condition $(\star\star\star)$ is checked; if the problem turns out to be unbounded, we terminate by returning the optimum value ∞. Finally, new auxiliary arcs are added in order to satisfy $(\star)$.

Removing arc capacities

Let us divide the arc set as $E' = E'_u \cup E'_\infty$, where $e \in E'_u$ if the capacity u'_e is finite, and $e \in E'_\infty$ if $u'_e = \infty$. Let the node set V consist of the original node set V' and a new node corresponding to every arc $e \in E'_u$; let $t := t'$. The original nodes are called *primary* nodes, and those corresponding to arcs *secondary* nodes. Let $k = a_{ij}$ be the node corresponding to arc $ij \in E'_u$. The transformed graph contains two corresponding arcs, ik and jk. We leave all arcs in $ij \in E'_\infty$ unchanged between the primary nodes i and j. Let us define $\bar{B}$ to be twice the product of the numerators and denominators of all rational numbers γ'_{ij} for every $ij \in E'$ and u'_{ij} for every $ij \in E'_u$; clearly, $\bar{B} \le 2B^{4m'}$.

For a primary node $i \in V$, let us set the node demand $b_i = -\sum_{j:ji \in E'_u} \gamma'_{ji} u'_{ji}$. For the secondary node $k = a_{ij}$, let $b_k := \gamma'_{ij} u'_{ij}$. Furthermore, let us define the gain factors by $\gamma_{ik} := \gamma'_{ij}$, $\gamma_{jk} := 1$. For $ij \in E'_\infty$, we let $\gamma_{ij} := \gamma'_{ij}$.

The transformed instance satisfies $(\star\star)$, since the following $\bar{f}$ is a feasible solution. For every secondary node $k = a_{ij}$, let us set $\bar{f}_{jk} := \gamma'_{ij} u'_{ij}$, and let us set $\bar{f}_{pq} = 0$ for all other arcs pq.

Boundedness

Let us now address the boundedness of the problem. The following lemma gives a simple characterization of boundedness of the objective.

Lemma 7.1. *Consider a problem instance (V, E, t, b, γ) in the uncapacitated formulation (P) that is feasible. The objective in (P) is bounded if and only if there is no cycle $C \subseteq E$ with $\gamma(C) > 1$ and a path $P \subseteq E$ between a node incident to C and t.*

Proof. If such a cycle exists, then we can increase the flow value in t arbitrarily by generating flow on C and sending it to t via P. For the converse direction, consider the dual program (D); recall that the labels μ_i are simply the inverses of the dual variables. Since (P) is feasible according to $(\star\star)$, the objective is bounded if and only if (D) is feasible. Assume $C \subseteq E$ is a cycle such that a path $P \subseteq E$ connects a node incident to C to t. Using the condition $\gamma_{pq}\mu_p/\mu_q \le 1$ on every arc $pq \in P$, it follows that in every feasible labeling, μ_i is finite for every node i incident to C. Therefore $\gamma(C) = \gamma^\mu(C) \le 1$, completing the proof. □

Let V' denote the set of nodes i such that there exists an $i - t$ path in E. This set V' can be found by a simple search algorithm. Boundedness can be decided by checking for a flow generating cycle in the restriction of G to V'. This is equivalent to finding a negative cycle for the cost function $c_{ij} = -\log \gamma_{ij}$ and can be solved by any negative cycle detection algorithm, see e.g. [2, Chapter 5.5]. Computations with logarithms can be avoided by devising a multiplicative analogue of these algorithms working directly with the γ_{ij}'s.

After removing the arc capacities, we run this algorithm to decide boundedness. If the problem is unbounded, we terminate with optimum value ∞. Otherwise, we can assume the validity of $(\star\star\star)$. (Note that since all secondary nodes have only two incoming arcs incident, all arcs used in C and P are necessarily from E_∞. Therefore, the same subroutine could also be performed before the transformation.)

Auxiliary arcs

To satisfy $(\star)$, for every node $i \in V - t$ for which $it \notin E$, let us further add an arc it to E with $\gamma_{it} := 1/\bar{B}$. Let us call these *auxiliary arcs.*

The following lemma justifies our transformation.

Lemma 7.2. *The transformed instance satisfies assumptions $(\star)$, $(\star\star)$ and $(\star\star\star)$, and $\bar{B}$ satisfies the assumptions on the encoding sizes in Section 2.1. An optimal solution f to the modified problem can be transformed to an optimal solution f' to the original problem in $O(m')$ time.*

Proof. The first part is straightforward. For the second statement, let f be an optimal solution to the modified problem with an optimal labeling μ as in Theorem 2.3(i). For a secondary node $k = a_{ij}$, let us set $f'_{ij} := f_{ik}$. Let $S_0 \subseteq V$ denote the set of nodes $i \in V$ for which $\gamma^{\mu}_{it} = 1$, that is $\mu_i = \bar{B}$. Let $S \subseteq V$ denote the set of nodes that can be reached from S_0 on a residual path $P \subseteq E_f$.

Let $S' \subseteq V'$ denote the set of primary nodes in S. Let us set $\mu'_i := \mu_i$ if $i \in V' \setminus S'$ and $\mu'_i := \infty$ if $i \in S'$. In what follows, we shall verify the optimality conditions in Theorem 2.3(ii) for f' and μ'.

We first claim that $f'_{ij} \leq u'_{ij}$ for all arcs $ij \in E'_u$. This follows since for the secondary node $k = a_{ij}$ we have $b_k = \gamma'_{ij} u'_{ij}$, and $e_k(f) = 0$ due to the optimality of f. Next, we claim that $t \notin S$ and therefore $\mu'_t = 1$. Indeed, assume for a contradiction there exists a path $P \subseteq E_f$ from a node $i \in S_0$ to t. Then $\mu_i \leq 1/\gamma(P) < \bar{B}$ by the definition of $\bar{B}$, a contradiction to $\mu_i = \bar{B}$.

The condition on arcs $ij \in E'[S']$ is straightforward since $\mu'_i = \mu'_j = \infty$. Consider an arc $ij \in E'$ with $i \in S'$, $j \in V' \setminus S'$. If $ij \in E'_\infty$, then $ij \in E_f$, contradicting the definition of S. Hence $ij \in E'_u$; let $k = a_{ij}$ be the corresponding secondary node. By definition, $ik \in E \subseteq E_f$. By the definition of S', we must have $kj \notin E_f$, that is, $f_{jk} = 0$ and therefore $f'_{ij} = u'_{ij}$ due to the constraint $e_k(f) = b_k$. Then $\gamma_{ij}\mu_i = \infty > \mu_j$, as required. It follows similarly that $f_{ij} = 0$ for all arcs $ij \in E'$ with $i \in V' \setminus S'$, $j \in S'$, and they satisfy $\gamma_{ij}\mu_i < \infty = \mu_j$.

Let us focus on arcs $ij \in E'[V' \setminus S']$; assume $ij \in E'_u$ and $0 < f'_{ij} < u'_{ij}$. This means that for the corresponding secondary node $k = a_{ij}$, we had $f_{ik}, f_{jk} > 0$, and thus $\gamma_{ij}\mu_i = \mu_k$, and $\mu_k = \mu_j$, implying $\gamma_{ij}\mu'_i = \mu'_j$. Note that $k \notin S_0$ and $e_k(f) = 0$ implies that $f_{ij} \leq u'_{ij}$, therefore $f'_{ij} = f_{ij}$ on all such arcs. The other cases, including the case of arcs in E'_∞, follow similarly.

It is left to prove that $e_i(f') = 0$ whenever $i \in V' \setminus S'$. By definition, $i \notin S_0$ and hence $f_{it} = 0$. For every incoming arc ji with secondary node $k = a_{ji}$, we have $f_{jk} = \gamma'_{ji}(u'_{ji} - f'_{ji})$. Together with $e_i(f) = 0$ and the definition of b_i, this implies $e_i(f') = 0$. □

7.2 Linear programs with two nonzeros per column

In this Section, we show how our algorithm can be used to solve arbitrary linear feasibility problems of the form (LP2).

The main part of this argument was given by Hochbaum [17], showing how an arbitrary instance of (LP2) can be transformed to another one where every column of the matrix A contains exactly one positive entry and a -1 entry. With the rows corresponding to nodes and the columns to arcs, let us use $\gamma_{ij} > 0$ to denote the positive entry in row j and column ij. (The construction creates two copies of the vertex set, and columns with two positive or two negative entries are represented by two arcs crossing between the copies, whereas columns with two different signs are represented by two arcs, one in each copy.) The transformed version may contain upper capacities on the arcs. These can be removed using the same construction as in Section 7.1, at the cost of increasing the number of nodes to $O(m)$. After removing the arc capacities, we can write the system in the form

$$\sum_{j:ji\in E}\gamma_{ji}f_{ji} - \sum_{j:ij\in E} f_{ij} = b_i \quad \forall i \in V \qquad (LP2M)$$
$$f \geq 0.$$

Given an instance of $(LP2M)$, let the value $\bar{B}$ be chosen as an integer multiple of the products of all numerators and denominators of the γ_{ij} values, and furthermore, assume $|b_i| \leq \bar{B}$ and b_i is an integer multiple of $1/\bar{B}$ for all $i \in V$.

$(LP2M)$ is an uncapacitated generalized flow feasibility problem, where all node demands must be exactly met (M stands for monotone, following Hochbaum's terminology.) Compared to the formulation (P), the differences are as follows: *(i)* $(LP2M)$ is a feasibility problem and does not have a distinguished sink node, in contrast to the optimization problem (P); *(ii)* the node demands must be exactly met in $(LP2M)$, whereas in (P), nodes are allowed to have excess. For this reason, we introduce two relaxations of $(LP2M)$ with inequalities.

$$\begin{aligned} \sum_{j:ji\in E}\gamma_{ji}f_{ji} - \sum_{j:ij\in E} f_{ij} &\geq b_i \quad \forall i\in V \\ f &\geq 0 \end{aligned} \tag{$LP2M_\geq$}$$

$$\begin{aligned} \sum_{j:ji\in E}\gamma_{ji}f_{ji} - \sum_{j:ij\in E} f_{ij} &\leq b_i \quad \forall i\in V \\ f &\geq 0 \end{aligned} \tag{$LP2M_\leq$}$$

Our main insight (Lemma 7.4 below) is that if both these relaxations are feasible, then $(LP2M)$ is also feasible, and a solution can be found efficiently provided the solutions to the relaxed instances.

The second relaxation $(LP2M_\leq)$ can be reduced to $(LP2M_\geq)$ by reversing all arcs in E, setting $\gamma_{ji} = 1/\gamma_{ij}$ on the reverse arc ji of $ij \in E$, and changing the node demands to $-b_i$. We show that $(LP2M_\geq)$ – and consequently, $(LP2M_\leq)$ – can be solved using our algorithm for (P).

Solving $(LP2M_\geq)$

As a preprocessing step, we identify the set Z of nodes that can be reached via a path in E from a flow generating cycle in E. That is, $i \in Z$ if there exists a cycle $C \subseteq E$, $\gamma(C) > 1$, and a path $P \subseteq E$ connecting a node of C to i. This set Z can be found efficiently using algorithms for negative cycle detection, similarly as in Section 7.1. Using the flow generating cycles, arbitrary demands b_i for $i \in Z$ can be met. This solves $(LP2M_\geq)$ if $Z = V$; in the sequel let us assume $V \setminus Z \neq \emptyset$. By the definition of Z, there is no arc in E between Z and $V \setminus Z$. If $(LP2M_\geq)$ is feasible, then there is a feasible solution with no arc carrying flow from $V \setminus Z$ to Z.

Thus we can reduce the problem to solving $(LP2M_\geq)$ on $V \setminus Z$. Let us add an artifical sink node t to V. For every $i \in V \setminus Z$ with $b_i > 0$, add a new arc ti with gain factor $\gamma_{ti} = 1$. For every $i \in V \setminus Z$, add an it arc with $\gamma_{it} = 1/\bar{B}$. This gives an instance of (P) with sink t. The condition $(\star)$ is guaranteed by the it arcs; for $(\star\star)$, we have a simple feasible solution: send b_i units of flow on γ_{ti} for every i with $b_i > 0$, and set the flow to 0 on all other arcs. The boundedness condition $(\star\star\star)$ is guaranteed by Lemma 7.1; note that by the definition of Z, there are no flow generating cycles in $E[V \setminus Z]$.

Lemma 7.3. *Let f be an optimal solution to the (P) instance as constructed above. Then $(LP2M_\geq)$ is feasible if and only if $f_{ti} = 0$ for all $i \in V \setminus Z$.*

Proof. Consider an optimal solution f to the instance of (P) with an optimal labeling μ. If $f_{ti} = 0$ for $i \in V \setminus Z$, then f restricted to $V \setminus Z$ is a feasible solution $(LP2M_\geq)$. Conversely, assume $f_{tj} > 0$ for a certain node $j \in V \setminus Z$; we show that $(LP2M_\geq)$ is infeasible.

As in the proof of Lemma 7.2, we let S_0 denote the set of nodes $i \in V \setminus Z$ with $\mu_i = \bar{B}$, and let S be the set of nodes that can be reached from S_0 on a residual path in E_f. We claim that $j \notin S$. To see this, first observe that $\mu_j = 1$ because of $f_{tj} > 0$. If there were a path $P \subseteq E_f$ from a node $i \in S$ to j, then $1 \geq \gamma^\mu(P) = \gamma(P)\mu_i/\mu_j = \gamma(P)\bar{B}$ gives a contradiction to the choice of $\bar{B}$.

Therefore $X = V \setminus (Z \cup S)$ contains j, and there is no arc entering this set. Further, $e_i(f) = 0$ and $f_{it} = 0$ for every $i \in X$. Then $y_i := 1/\mu_i$ for $i \in X$ and $y_i := 0$ for $i \notin X$ gives a Farkas certificate of infeasibility for $(LP2M_\geq)$.[3] Indeed, $y \geq 0$, $y_i - y_j\gamma_{ij} \geq 0$ holds for every arc $ij \in E$,

[3] The Farkas certificate is described after the proof of Lemma 7.4.

and $\sum_{i\in V} b_i y_i > 0$ because

$$\sum_{i\in V} b_i y_i = \sum_{i\in X} b_i^\mu = \sum_{i\in X}\sum_{j\in V\cup\{t\}:ji\in E} \gamma_{ji}^\mu f_{ji}^\mu - \sum_{j\in V:ji\in E} f_{ji}^\mu = \sum_{i\in V} f_{ti}^\mu > 0,$$

completing the proof. Here we used that $\gamma_{ji}^\mu = 1$ whenever $f_{ji} > 0$. □

Solving $(LP2M)$

We solve $(LP2M_\geq)$ as described above, and $(LP2M_\leq)$ the same way, after reversing the arcs. If either of the two problems is infeasible, then $(LP2M)$ is also infeasible. Assume now that f is a feasible solution to $(LP2M_\geq)$, and g is a feasible solution to $(LP2M_\leq)$. We show that in this case the equality version $(LP2M)$ is also feasible. To prove this, we use a flow decomposition of the difference of the two solutions f and g to transform g to a solution of $(LP2M_\leq)$.

Lemma 7.4. *Given feasible solutions to $(LP2M_\geq)$ and $(LP2M_\leq)$, a feasible solution to $(LP2M)$ can be found in strongly polynomial time.*

Proof. Let f be a feasible solution to $(LP2M_\geq)$, and g a feasible solution to $(LP2M_\leq)$. Then for every $i \in V$, $e_i(f) \geq 0 \geq e_i(g)$ holds. Let us define the flow h as

$$h_{ij} := \begin{cases} f_{ij} - g_{ij} & \text{if } ij \in E,\ f_{ij} > g_{ij} \\ \gamma_{ji}(f_{ji} - g_{ji}) & \text{if } ji \in E,\ f_{ji} > g_{ji}. \end{cases}$$

Let $H \subseteq \overleftrightarrow{E}$ denote the support of h; clearly, $h_{ij} > 0$ for every $ij \in H$. With the convention $h_{ij} = -\gamma_{ji} h_{ji}$, we have $f = g + h$. Since $e_i(f) \geq 0 \geq e_i(g)$, the inequality $\sum_{j:ji\in H} \gamma_{ji} h_{ji} \geq \sum_{j:ij\in H} h_{ij}$ holds for every $i \in V$.

We apply the standard generalized flow decomposition for h as in e.g. [15, 9]: every generalized flow can be written as the sum of five types of elementary flows. Such a decomposition can be found in $O(nm)$ time, and the number of terms is at most the number of arcs with positive flow.

Among the five types of elementary flows listed in [9], Types I and III cannot be present the decomposition of h, as there are no deficit nodes (more outgoing than incoming flow). Type IV are unit gain cycles, and Type V are pairs of flow generating and flow absorbing cycles connected by a path ("bicycles"); these do not generate any excess or deficit and are not needed for out argument. The important one is Type II: a flow generating cycle and a path connecting it to an excess node (more incoming than outgoing flow).

We now describe how to modify g to a feasible solution g' to $(LP2M)$ using the decomposition of h. Consider a node i with $e_i(f) > e_i(g)$; this is an excess node for h. We could add all Type II flows in the decomposition terminating at i to increase $e_i(g)$ to $e_i(f)$. Since we want achieve the equality $e_i(g') = 0$, we only use some of the Type II flows. We add them one-by-one until $e_i(g')$ becomes nonnegative. Then for the last flow, we add only a fractional amount to set precisely $e_i(g') = 0$. Repeating this for every i with $e_i(f) > e_i(g)$, we obtain a feasible solution g' to $(LP2M)$. □

We also present a second proof of the claim that if both $(LP2M_\geq)$ and $(LP2M_\leq)$ are feasible, then $(LP2M)$ is also feasible. The proof is based on Farkas's lemma and is not algorithmic, but may contribute to a better understanding of the claim.

We show that if $(LP2M)$ is infeasible, then either $(LP2M_\geq)$ or $(LP2M_\leq)$ is also infeasible. A Farkas-certificate to the infeasibility of $(LP2M)$ can be written as

$$\begin{aligned} y_i - y_j\gamma_{ij} &\geq 0 \quad \forall ij \in E \\ \sum_{i\in V} b_i y_i &> 0 \end{aligned}$$

A Farkas-certificate to the infeasibility $(LP2M_{\geq})$ is the same with the additional constraint $y \geq 0$, whereas the certificate to the infeasibility of $(LP2M_{\leq})$ is with $y \leq 0$. In the case of $(LP2M_{\geq})$, $\mu_i = 1/y_i$ gives the usual labeling.

Let us define the sets $Y^+ := \{i \in V : y_i > 0\}$ and $Y^- := \{i \in V : y_i < 0\}$. Further, let $y_i^+ := y_i$ if $i \in Y^+$ and $y_i^+ := 0$ otherwise; similarly, let $y_i^- := y_i$ if $i \in Y^-$ and 0 outside Y^-.

We claim that the $y_i^+ - y_j^+\gamma_{ij} \geq 0$ and $y_i^- - y_j^-\gamma_{ij} \geq 0$ hold for every $ij \in E$. We only verify this for y^+; the proof is the same for y^-. If $i, j \in Y^+$, then this holds because y^+ is identical to y inside Y^+. If $i, j \in V \setminus Y^+$, then $y_i^+ = y_j^+ = 0$ and thus the claim is trivial. Next, let $i \in Y^+$ and $j \in V \setminus Y^+$. The claim follows by $y_i^+ > 0$, and $y_j^+ = 0$. Finally, we claim that there is no $ij \in E$ with $i \in V \setminus Y^+$, $j \in Y^+$. Indeed, this would mean $y_i - y_j\gamma_{ij} < 0$, contradicting the choice of y.

Since $0 < \sum_{i\in V} b_i y_i = \sum_{i\in V} b_i y_i^+ + \sum_{i\in V} b_i y_i^-$, either $\sum_{i\in V} b_i y_i^+ > 0$ or $\sum_{i\in V} b_i y_i^- > 0$. In the first case, y^+ is an infeasibility certificate for $(LP2M_{\geq})$, and in the second case, y^- is an infeasibility certificate for $(LP2M_{\leq})$.

8 Conclusion

We have given a strongly polynomial algorithm for the generalized flow maximization problem, and also for solving feasibility LPs with at most two nonzero entries in every column of the constraint matrix. A natural next question is to address the minimum cost generalized flows, or equivalently, finding optimal solutions to LPs with two nonzero entries per column.

In contrast to the vast literature on the flow maximization problem, there is only one weakly polynomial combinatorial algorithm known for this setting, the one by Wayne [35]. This setting is more challenging since the dual structure cannot be characterized via the convenient relabeling framework, and thereby most tools for minimum cost circulations, including the scaling approach also used in this paper, become difficult if not impossible to apply.

Another possible line of research would be to extend the flow maximization algorithm to nonlinear settings. The paper [34] gave a simple scaling algorithm for *concave generalized flows*, where instead of the gain factors γ_e, there is a concave increasing function $\Gamma_e(.)$ associated to every arc e. In [33], a strongly polynomial algorithm is given to the analogous problem of minimum cost circulations with separable convex cost functions satisfying certain assumptions. One could combine the techniques of [34] and [33] with the ideas of the current paper to obtain strongly polynomial algorithms for some special classes of concave generalized flow problems. This could also lead to strongly polynomial algorithms for certain market equilibrium computation problems, see [34].

Acknowledgment

The author is grateful to Joseph Cheriyan, Ian Post, and the anonymous referees for several suggestions that helped to improve the presentation.

References

[1] I. Adler and S. Cosares. A strongly polynomial algorithm for a special class of linear programs. *Operations Research*, 39(6):955–960, 1991.

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

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

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

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

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

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

[8] F. Glover and D. Klingman. On the equivalence of some generalized network problems to pure network problems. *Mathematical Programming*, 4(1):269–278, 1973.

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

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

[11] D. Goldfarb and Z. Jin. A faster combinatorial algorithm for the generalized circulation problem. *Mathematics of Operations Research*, 21(3):529–539, 1996.

[12] D. Goldfarb, Z. Jin, and Y. Lin. A polynomial dual simplex algorithm for the generalized circulation problem. *Mathematical Programming*, 91(2):271–288, 2002.

[13] D. Goldfarb, Z. Jin, and J. B. Orlin. Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem. *Mathematics of Operations Research*, 22(4):793–802, 1997.

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

[15] M. Gondran and M. Minoux. *Graphs and Algorithms.* John Wiley and Sons, NY, USA, 1984.

[16] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 1993.

[17] D. S. Hochbaum. Monotonizing linear programs with up to two nonzeroes per column. *Operations Research Letters*, 32(1):49–58, 2004.

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

[19] L. V. Kantorovich. Mathematical methods of organizing and planning production. *Publication House of the Leningrad State University*, page 68, 1939. English translation in *Management Science* 6(4):366-422, 1960.

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

[21] N. Megiddo. Towards a genuinely polynomial algorithm for linear programming. *SIAM Journal on Computing*, 12(2):347–353, 1983.

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

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

[24] J. B. Orlin. Max flows in $O(nm)$ time or better. In *Proceedings of STOC*, 2013.

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

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

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

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

[29] É. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985.

[30] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, pages 250–256, 1986.

[31] É. Tardos and K. D. Wayne. Simple maximum flow algorithms in lossy networks. In *Proceedings of IPCO, Lecture Notes in Computer Science*, volume 1412, pages 310–324, 1998.

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

[33] L. A. Végh. Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In *Proceedings of STOC*, pages 27–40. ACM, 2012.

[34] L. A. Végh. Concave generalized flows with applications to market equilibria. *Mathematics of Operations Research*, 39(2):573–596, 2014.

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

Appendix

Theorem 4.1. *Let (f, μ) be a Δ-feasible pair. Then there exists an optimal solution f^* such that*

$$||f^\mu - f^{*\mu}||_\infty \le Ex^\mu(f) + (|F^\mu| + 1)\Delta.$$

Proof. First, let us modify (f,μ) to a conservative pair $(\tilde f,\mu)$ by setting the flow values on non-tight arcs to 0, as in Lemma 2.4. We shall prove the existence of an optimal f^* such that

$$||\tilde f^\mu - f^{*\mu}||_\infty \le Ex^\mu(\tilde f). \tag{12}$$

This implies the claim, since Lemma 2.4 asserts $Ex^\mu(\tilde f) \le Ex^\mu(f) + |F^\mu_f|\Delta$, and $||\tilde f^\mu - f^\mu||_\infty \le \Delta$ as the two flows differ only on non-tight arcs.

Let us pick an optimal solution f^* to (P) such that $||\tilde f - f^*||_1$ is minimal, and let μ^* be an optimal solution to (D). Note that because of $(\star)$, all values of μ and μ^* are finite. We use a similar argument as in the proof of Lemma 7.4. Let us define

$$h_{ij} := \begin{cases} f^*_{ij} - \tilde f_{ij} & \text{if } ij \in E,\ f^*_{ij} > \tilde f_{ij} \\ \gamma_{ji}(f^*_{ji} - \tilde f_{ji}) & \text{if } ji \in E,\ f^*_{ji} > \tilde f_{ji}. \end{cases}$$

Let $H \subseteq \overleftrightarrow{E}$ denote the support of h; clearly, $h > 0$ and $H \subseteq E_{\tilde f}$ whereas $\overleftarrow{H} \subseteq E_{f^*}$. Again, with the convention $h_{ij} = -\gamma_{ji}h_{ji}$, we have $f^* = \tilde f + h$.

Claim 8.1. *The arc set H does not contain any directed cycles.*

Proof. First, let $C \subseteq H$ be a cycle. Since μ is a conservative labeling for $\tilde f$ and $C \subseteq E_{\tilde f}$, we have $\gamma(C) = \gamma^\mu(C) \le 1$. On the other hand, μ^* is conservative for f^* and $\overleftarrow{C} \subseteq E_{f^*}$. Therefore $\gamma(\overleftarrow{C}) = 1/\gamma(C) = 1/\gamma^{\mu^*}(C) \le 1$. These together give $\gamma(C) = \gamma(\overleftarrow{C}) = 1$, and also $\gamma^{\mu^*}_e = 1$ for every $e \in C$. Hence we can modify f^* to another optimal solution by decreasing every $f^{*\mu^*}_e$ value by a small $\varepsilon > 0$. This gives a contradiction to our extremal choice of f^* as the optimal solution minimizing $||\tilde f - f^*||_1$. □

Observe that

$$e_i(\tilde f) - e_i(f^*) = \sum_{j:ij\in H} h_{ij} - \sum_{j:ji\in H} \gamma_{ji}h_{ji}$$

By the optimality of f^*, the left hand side is ≤ 0 for $i = t$ and is equal to $e_i(\tilde f) \ge 0$ otherwise. The above claim guarantees that H, the support of h, is acyclic. Consequently, we can easily decompose h to the form

$$h = \sum_{1\le\ell\le k} h^\ell,$$

where each h^ℓ is a path flow with support P^ℓ from a node p^ℓ with $e_{p^\ell}(\tilde f) > 0$ to t, and $k \le m$.

Such a decomposition is easy to construct by using a topological order of the nodes for H. It is also a special case of the flow decomposition argument used in Lemma 7.4, see also [15, 9]. (The difference is that according to Claim 8.1, four out of the five types of elementary flows, Types II-V cannot exist as they contain cycles.)

Let λ^ℓ denote the value of h^ℓ on the first arc of P^ℓ. Since μ is a conservative labeling and $P^\ell \subseteq H \subseteq E_{\tilde f}$, we have $\gamma^\mu_{ij} \le 1$ for all arcs of P^ℓ and therefore the relabeled flow $(h^\ell)^\mu$ is monotone decreasing along P^ℓ. Hence it follows that for every arc ij,

$$h^\mu_{ij} = \sum_{1\le\ell\le k} (h^\ell_{ij})^\mu \le \sum_{1\le\ell\le k} \frac{\lambda^\ell}{\mu_{p^\ell}} = \sum_{i:V-t} e^\mu_i(\tilde f) = Ex^\mu(\tilde f).$$

This completes the proof, since $||\tilde f^\mu - f^{*\mu}||_\infty = \max_{ij\in E} h^\mu_{ij}$ (note that if $f^*_{ij} < f_{ij}$, then $\gamma^\mu_{ij} = 1$ must hold). □