On Weighted Multicommodity Flows in Directed Networks

Maxim A. Babenko* Alexander V. Karzanov†

Abstract

Let $G = (VG, AG)$ be a directed graph with a set $S \subseteq VG$ of terminals and nonnegative integer arc capacities c. A feasible *multiflow* is a nonnegative real function $F(P)$ of "flows" on paths P connecting distinct terminals such that the sum of flows through each arc a does not exceed $c(a)$. Given $\mu\colon S \times S \to \mathbb{R}_+$, the *$\mu$-value* of F is $\sum_P F(P)\mu(s_P, t_P)$, where s_P and t_P are the start and end vertices of a path P, respectively.

Using a sophisticated topological approach, Hirai and Koichi showed that the maximum μ-value multiflow problem has an integer optimal solution when μ is the distance generated by subtrees of a weighted directed tree and (G, S, c) satisfies certain Eulerian conditions.

We give a combinatorial proof of that result and devise a strongly polynomial combinatorial algorithm.

Keywords: directed multiflow, tree-induced distance, strongly polynomial algorithm

1 Introduction

1.1 Multiflows in directed networks

We use standard terminology of graph and flow theory. For a digraph G, the sets of its vertices and arcs are denoted by VG and AG, respectively. A similar notation is used for paths, cycles, and etc. For $X \in VG$, the set of arcs of G entering (resp. leaving) X is denoted by $\delta^{\rm in}_G(X)$ (resp. $\delta^{\rm out}_G(X)$). When $X = \{v\}$, we write $\delta^{\bullet}_G(v)$ for $\delta^{\bullet}_G(\{v\})$. When G is clear from the context, it is omitted from notation. Also for a set A and a singleton a, we will write $A - a$ for $A \setminus \{a\}$, and $A \cup a$ for $A \cup \{a\}$.

*Moscow State University; Leninskie Gory, 119991 Moscow, Russia; email: maxim.babenko@gmail.com. A part of this research was done while this author was visiting Institut für Diskrete Mathematik, Univ. Bonn.

†Inst. for System Analysis of the RAS; 9, Prospect 60 Let Oktyabrya, 117312 Moscow, Russia; email: sasha@cs.isa.ac.ru. Supported by RFBR grant 10-01-9311-CNRSL_ a.

A *directed network* is a triple $\mathcal{N} = (G, S, c)$ consisting of a digraph G, a set $S \subseteq VG$ of *terminals*, and *integer* arc *capacities* $c\colon AG \to \mathbb{Z}_+$. Vertices in $VG - S$ are called *inner*. A directed path in G is called an S*-path* if its endvertices are distinct elements of S. A *multiflow* F is a function assigning a nonnegative real number, or *flow*, to each S-path. A multiflow F is called *feasible* if for each arc $a \in AG$, the sum of flows assigned to S-paths going through a does not exceed $c(a)$. The (total) *value* of F is the sum of flows over all S-paths P:

$$\text{val}(F) := \sum\nolimits_P F(P). \tag{1.1}$$

Sometimes (e.g., in [IKN98]) such multiflows are called *free* to emphasize the fact that *any* pair of terminals is allowed to be connected by nonzero flows.

The following *maximum (fractional) multiflow problem* is well known:

(MF) *Given a directed network* $\mathcal{N} = (G, S, c)$*, find a feasible multiflow* F *of maximum value.*

The problem in which one is asked for maximizing among the *integer* multiflows is denoted by **IMF**.

For general directed networks $\mathcal{N}$, problem **IMF** is NP-hard already for $|S| = 2$ [FHW80]. Tractable cases have been revealed for networks obeying a certain conservation property. More precisely, c (or $\mathcal{N}$) is called *Eulerian* at a vertex v if $c(\delta^{\text{in}}(v)) = c(\delta^{\text{out}}(v))$. (For a function $f : A \to \mathbb{R}$ and a subset $A' \subseteq A$, we write $f(A')$ for $\sum(f(a)\colon a \in A')$.) When c is Eulerian at all inner vertices (resp. at all vertices), the network $\mathcal{N}$ is called *inner* (resp. *totally*) *Eulerian*. By a *cut* in G we mean a pair of nonempty subsets $(X, \overline{X})$, where $X \subset VG$ and $\overline{X} := VG - X$. It is called an (S_1, S_2)*-cut* if $S_1 \subseteq X$ and $S_2 \subseteq \overline{X}$. When it is not confusing, we may refer to the arc sets $\delta^{\text{out}}_G(X)$ and $\delta^{\text{in}}_G(X)$ as cuts as well.

The following result signifies the importance of inner Eulerian networks:

Theorem 1.1 (Lomonosov (unpublished, 1978), Frank [Fr89]) *Let* $\mathcal{N} = (G, S, c)$ *be an inner Eulerian directed network. Then there exists an integer maximum feasible multiflow* F *in* $\mathcal{N}$*. It satisfies*

$$\text{val}(F) = \sum\nolimits_{t \in S} c(\delta^{\text{out}}(X_t)),$$

where for each $t \in S$, $(X_t, \overline{X}_t)$ *is a minimum capacity* $(t, S - t)$*-cut in* $\mathcal{N}$*.*

Therefore, **MF** and **IMF** have the same optimal value for an inner Eulerian network, and this value can be found in strongly polynomial time (by computing a minimum $(t, S - t)$-cut for each $t \in S$). Ibaraki, Karzanov, and Nagamochi [IKN98] devised a "divide-and-conquer" method that computes an integer maximum multiflow in such a network in $O((MF(n, m)+mn)\cdot\log|S|+mn^2)$ time. (Hereinafter $n := |VG|$, $m := |AG|$, and $MF(n', m')$ denotes the complexity of a max-flow computation in a directed network with n' vertices and m' arcs.) The latter complexity was improved to $O((MF(n, m) + mn \log(n^2/m)) \cdot \log|S|)$ in [BK07].

1.2 Weighted multiflows

A generalization of **MF** involves weights between terminals. More precisely, given a *weighting* $\mu\colon S\times S\to\mathbb{R}_+$, the *$\mu$-value* of a multiflow F is

$$\text{val}(F,\mu) := \sum\nolimits_P \mu(s_P,t_P)F(P), \tag{1.2}$$

where the sum is over all S-paths P, and s_P and t_P denote the start and end vertices of P, respectively. We may assume that $\mu(s,s)=0$ for all $s\in S$.

Replacing (1.1) by (1.2), we obtain the weighted counterpart of **MF**:

(μ-MF) *Given $\mathcal{N}$ and μ as above, find a feasible multiflow F of maximum μ-value.*

The integer strengthening of μ-**MF** is denoted by μ-**IMF**. When $\mu(s,t)=1$ for all $s\neq t$, μ-**MF** turns into **MF**, and μ-**IMF** into **IMF**.

1.3 Tree-induced weights

It has been shown that problem μ-**IMF** has a rather wide spectrum of tractable cases. The simplest case is $S=\{s,t\}$, $\mu(s,t)=1$ and $\mu(t,s)=0$; then μ-**IMF** becomes the standard maximum flow problem with arbitrary integer capacities. A representative well-solvable class has been found in connection with the so-called *directed multiflow locking problem*:

(DMLP) *Given a directed network $\mathcal{N}=(G,S,c)$ and a collection $\mathcal{C}\subseteq 2^S$, find a feasible multiflow F in $\mathcal{N}$ that locks simultaneously all members of $\mathcal{C}$.*

Here F is said to *lock* a subset $A\subset S$ if the sum of values $F(P)$ over the S-paths P going from A to $S-A$ is maximum possible, i.e. it is equal to the minimum capacity of an $(A,S-A)$-cut in $\mathcal{N}$. A collection $\mathcal{C}\subseteq 2^S$ is called *lockable* if **DMLP** has a solution for all (G,c) (with S fixed). Important facts are given in the following

Theorem 1.2 ([IKN98]) *$\mathcal{C}\subseteq 2^S$ is lockable if and only if $\mathcal{C}$ is* cross-free*, i.e. for any $A,B\in\mathcal{C}$, at least one of the following holds: $A\subseteq B$, $B\subseteq A$, $A\cap B=\emptyset$, $A\cup B=S$. Moreover, if $\mathcal{C}$ is cross-free and $\mathcal{N}$ is inner Eulerian, then **DMLP** has an integer solution. Such a solution can be found in $O((MF(n,m)+mn)\cdot\log|S|+mn^2)$ time.*

This gives rise to the following tractable cases of μ-**IMF**. Given $\mathcal{C}\subseteq 2^S$, take an arbitrary function $\ell:\mathcal{C}\to\mathbb{R}_+$. For $s,t\in S$, define

$$\mu_\ell(s,t) := \sum(\ell(A)\colon A\in\mathcal{C},\ s\in A\not\ni t). \tag{1.3}$$

Suppose that $\mathcal{C}$ is cross-free and $\mathcal{N}$ is inner Eulerian, and let F be an integer solution to **DMLP** (existing by Theorem 1.2). Then F is simultaneously an optimal solution to μ_ℓ-**IMF** for every $\ell:\mathcal{C}\to\mathbb{R}_+$; this can be easily concluded from the fact that F saturates minimum capacity $(A,S-A)$-cuts in $\mathcal{N}$ for all $A\in\mathcal{C}$.

A cross-free collection $\mathcal{C}$ can be represented by use of a directed tree $T=(VT,AT)$ (a digraph whose underlying undirected graph is a tree); namely:

(1.4) there is a bijection $\beta : \mathcal{C} \to AT$ and a map $\gamma : S \to VT$ such that for each arc $a = (u, v) \in AT$ and for $A := \beta^{-1}(a)$, the set of terminals $s \in S$ whose image $\gamma(s)$ occurs in the component of $T - a$ containing u is exactly A.

Under this correspondence, we may interpret $\ell : \mathcal{C} \to \mathbb{R}_+$ as a *length* function on the arcs of T, keeping the same notation: $\ell(a) := \ell(A)$ for $a \in AT$ and $A := \beta^{-1}(a)$. These arc lengths induce *distances* $d = d_\ell$ on VT in a natural way:

(1.5) for $x, y \in VT$, define $d(x, y)$ to be the sum of ℓ-lengths of *forward* arcs in the simple path from x to y in T.

(This path may contain both forward and backward arcs. If there are no forward arcs, we have $d(x, y) = 0$.) One can see that $\mu = \mu_\ell$ figured in (1.3) satisfy

(1.6) $$\mu(s, t) = d_\ell(\gamma(s), \gamma(t)) \qquad s, t \in S.$$

Note that such a μ satisfies the triangle inequalities $\mu(s, t) + \mu(t, u) \geq \mu(s, u)$ for all $s, t, u \in S$, i.e. μ is a *directed metric (space)*. In light of (1.5)–(1.6), μ is called a *tree-induced* directed metric.

Generalizing the above-mentioned integrality results, Hirai and Koichi [HK10] considered arbitrary weight (or *distance*) functions $\mu : S \times S \to \mathbb{R}_+$ and gave an exhaustive analysis of the integrality and "unbounded fractionality" behavior of problem μ-**MF** in terms of μ, for both integer and Eulerian cases.

More precisely, let T be a directed tree with nonnegative arc lengths ℓ. Given a network $\mathcal{N}$ as before, suppose that each terminal $s \in S$ is associated with some *subtree* (a weakly connected subgraph) T_s of T. For $s, t \in S$, define $\mu(s, t)$ to be the distance from T_s to T_t, i.e.

$$\mu(s, t) := \min\{d(u, v) \colon u \in T_s,\ v \in T_t\}.$$

(Note that such a μ need not satisfy triangle inequalities. When each T_s is a single vertex, μ is specified as in (1.6).)

A weight function μ on $S \times S$ that can be obtained in this way is called a *tree-induced (directed) distance*, and an appropriate tuple $\mathcal{R} = (T, \ell, \{T_s\})$ is called a *(tree) realization* of μ.

For such an $\mathcal{R}$, we distinguish between three sorts of terminals. We call $s \in S$ *simple* if T_s consists of a single vertex, *linear* if T_s is a directed path, and *complex* otherwise. If the whole tree T forms a directed path, then $\mathcal{R}$ itself is called *linear*.

Hirai and Koichi obtained the following results.

Theorem 1.3 ([HK10]) *Let $\mathcal{N} = (G, S, c)$ be a directed network and let $\mu\colon S \times S \to \mathbb{R}_+$ be a tree-induced distance with a realization $\mathcal{R}$.*

(i) *If $\mathcal{R}$ is linear, then μ-**MF** has an integer optimal solution.*

(ii) *If c is Eulerian at all inner vertices and all complex terminals, then μ-**MF** has an integer optimal solution.*

Note that the proof of (i) in Theorem 1.3 given in [HK10] is constructive; it reduces problem μ-**MF** in this case to finding a certain min-cost circulation. The proof in [HK10] is much more involved; it employs a topological approach based on a concept of tight spans of directed distance spaces introduced in that paper. (Another nice result in [HK10] relying on the directed tight span approach asserts that if a distance μ is not tree-induced, then μ-**MF** has unbounded fractionality in the totally Eulerian case, i.e. there is no positive integer k such that μ-**MF** admits a $\frac{1}{k}$-integer solution for every totally Eulerian network and this μ.)

In this paper we devise an efficient combinatorial algorithm that constructs an integer optimal solution to problem μ-**MF** under the conditions as in (ii) of Theorem 1.3; this yields an alternative (and relatively simple) proof of assertion (ii). Our method extends the divide-and-conquer approach of [IKN98]; it is described in Section 2. The algorithm runs in $O((MF(n,m) + mn\log(n^2/m)) \cdot \log|S|)$ time.

2 Algorithm

Let $\mathcal{N} = (G, S, c)$ and μ obey the conditions in Theorem 1.3(ii).

The following convention will allow us to slightly simplify the description of our algorithm (without loss of generality). In a tree realization $\mathcal{R} = (T, \ell, \{T_s\})$ of a distance function μ, let us think of T as an *undirected* tree with *edge* set ET, and assume that each edge $e = uv$ generates two oppositely directed arcs: one going from u to v, and the other from v to u (yielding a "directed quasi-tree"). For $a = (u, v)$, the opposite arc (v, u) is denoted by $\overline{a}$. The length function ℓ is given on the corresponding arc set, denoted by AT as before, and for $x, y \in VT$, the distance $d(x, y)$ is defined to be the ℓ-length of the corresponding *directed* path from x to y. Accordingly, a terminal s is linear if the subtree T_s is a simple undirected path and one of the two directed paths behind T_s has zero ℓ-length.

Sometimes, to ensure the desired efficiency of the method, we will be forced to treat some linear terminals as complex ones (which will never be confusing).

2.1 Initial reductions

Let $\mathcal{R} = (T, \ell, \{T_s\})$ be a tree realization of μ. A *pre-processing stage* of the algorithm applies certain reductions to $\mathcal{R}$ (called *initial reductions*).

Choose a linear terminal $s \in S$ (if exists), i.e. T_s is a path in T connecting some vertices t_1 and t_2, and one of the two directed paths behind T_s, from t_2 to t_1 say, has zero ℓ-length. This implies that

$$(2.1) \qquad \mu(x, s) = d(T_x, t_1) \quad \text{and} \quad \mu(s, x) = d(t_2, T_x) \qquad \text{for all } x \in S.$$

We replace s by a pair of simple terminals as follows. Add to G new terminals s_1 and s_2 and arcs (s, s_1) and (s_2, s). The capacities of these arcs are chosen to be sufficiently large and to make the network Eulerian at s. Denote the resulting digraph by G' and the network by $\mathcal{N}' := (G', S', c')$, where $S' := (S - s) \cup \{s_1, s_2\}$. We modify $\mathcal{R}$ into the tuple $\mathcal{R}'$ with the same tree T by setting $T_{s_1} := \{t_1\}$ and $T_{s_2} := \{t_2\}$. This gives new distance $\mu' \colon S' \times S' \to \mathbb{R}_+$.

We claim that the two problems: μ-**MF** with $(\mathcal{N}, \mathcal{R})$ and μ'-**MF** with $(\mathcal{N}', \mathcal{R}')$, are essentially equivalent. Indeed, $\mu'(s_2, s_1) = 0$ (since the ℓ-length of the directed path from t_2 to t_1 in T is zero). Therefore, one may consider only those multiflows in $\mathcal{N}'$ that are zero on all s_2–s_1 paths. Any other S'-path P' in $\mathcal{N}'$ has a natural image (an S-path) P in $\mathcal{N}$. Namely, if P' neither starts at s_2 nor ends at s_1, then $P = P'$. If P' starts at s_2 (resp. ends at s_1), then P is its maximal subpath from s (resp. to s). This gives a one-to-one correspondence between the S'-paths P' in $\mathcal{N}'$ (excluding s_2–s_1 ones) and the S-paths P in $\mathcal{N}$, and by (2.1), the transformation preserves distances: $\mu'(s_{P'}, t_{P'}) = \mu(s_P, t_P)$. We reset $\mathcal{N} := \mathcal{N}'$ and $\mathcal{R} := \mathcal{R}'$.

Making a sequence of similar reductions, we obtain a situation when

(C1) Any terminal in $\mathcal{N}$ is either simple or complex.

Four more sorts of reductions are applied to ensure the following additional properties:

(C2) Each leaf in T corresponds to some (possibly multiple) simple terminal. (For otherwise the leaf can be removed from T.)

(C3) No inner vertex v of T corresponds to a simple terminal. (For otherwise one can add to T a new vertex v' and edge vv' with zero ℓ-length of both arcs $(v, v'), (v', v)$, and replace the subtree $\{v\}$ by $\{v'\}$ in the realization of μ.)

(C4) Each inner vertex of T has degree at most 3. (This can be achieved by splitting inner vertices of bigger degrees in T and by adding additional edges with zero ℓ-length of arcs in both directions.)

(C5) T has $O(|S|^2)$ vertices.

To provide (C5), note that T has $O(|S|)$ leaves (by (C2)), and hence it has $O(|S|)$ vertices of degree 3. Consider a vertex v of degree 2 in T, and let $e = uv$ and $e' = vw$ be its incident edges. If for any $s \in S$, the subtree T_s contains either none or both of e, e', then we can merge e, e' into one edge uw (adding up the corresponding arc lengths), obtaining a realization of μ with a smaller tree size. Otherwise v is a leaf of some T_s. Obviously, the number of leaves of T_s does not exceed that of T, so it is estimated as $O(|S|)$. This gives (C5).

2.2 Optimality certificate

Here we establish a sufficient condition that implies optimality of a given multiflow. We need some additional terminology and notation. A feasible multiflow F in $\mathcal{N}$ is said to *saturate* a cut $(X, \overline{X})$ in G if each S-path P with $F(P) > 0$ meets $\delta^{\rm in}(X) \cup \delta^{\rm out}(X)$ at most once, and

$$\sum\nolimits_{P\colon e \in AP} F(P) = c(e) \quad \text{for each arc } e \in \delta^{\rm out}(X).$$

Definition. For an arc $a = (u, v)$ of T, define Π_a to be the set of pairs (s, t) in S such that $\mu(s, t)$ "feels" $\ell(a)$, i.e. a belongs to a minimal directed path that starts in T_s and ends in T_t. (Then $\mu(s, t)$ is just the ℓ-length of such path.)

For a multiflow F and a set $\Pi \subseteq S \times S$, let $F[\Pi]$ be the "restriction" of F relative to Π. More precisely, for an S-path P in G, we define

$$F[\Pi](P) := \begin{cases} F(P) & \text{if } (s_P, t_P) \in \Pi, \\ 0 & \text{otherwise.} \end{cases}$$

A cut $(X, \overline{X})$ in G is called Π-*separating* if $s \in X \not\ni t$ holds for each $(s, t) \in \Pi$.

Lemma 2.1 *Let F be a feasible multiflow in $\mathcal{N}$. Suppose that*

(2.2) there exists a collection $\{(X_a, \overline{X}_a)\colon a \in AT\}$ of cuts in G such that for each $a \in AT$, $(X_a, \overline{X}_a)$ is a Π_a-separating cut saturated by F.

*Then F is an optimal solution to μ-**MF**.*

Proof For $s, t \in S$, let $f(s, t)$ denote the sum of flows (by F) over the paths from s to t in G. Then

$$\mathrm{val}(F, \mu) = \sum\nolimits_{(s,t) \in S \times S} f(s, t)\mu(s, t).$$

Also

$$\mu(s, t) = \sum\nolimits_{a \in AT\colon (s,t) \in \Pi_a} \ell(a).$$

It follows that

$$\mathrm{val}(F, \mu) = \sum\nolimits_{a \in AT} \ell(a) \left(\sum\nolimits_{(s,t) \in \Pi_a} f(s, t) \right). \tag{2.3}$$

Consider an arc $a \in AT$. Since $(X_a, \overline{X}_a)$ is a Π_a-separating cut, we have

$$\sum\nolimits_{(s,t) \in \Pi_a} f(s, t) \le c(\delta^{\rm out}(X_a)), \tag{2.4}$$

Then (2.3) and (2.4) give

$$\mathrm{val}(F, \mu) \le \sum\nolimits_{a \in AT} \ell(a)\, c(\delta^{\rm out}(X_a)). \tag{2.5}$$

Since each cut $(X_a, \overline{X}_a)$ is saturated by F, inequality (2.4) turns into equality, and so does (2.5). Thus, $\mathrm{val}(F, \mu)$ is maximum, and the lemma follows. □

Given a problem instance $(\mathcal{N}, \mathcal{R})$, our algorithm will construct an integer multiflow F that possesses property (2.2), and therefore F is optimal by Lemma 2.1. Note that (2.2) does not involve the lengths ℓ of arcs in T, so F is optimal simultaneously for all distances μ induced by arbitrary ℓ (when T and $\{T_s\}$ are fixed).

2.3 Partitioning step

The core of the algorithm consists in the following recursive procedure that divides the current instance $(\mathcal{N}, \mathcal{R})$ into a pair of smaller ones.

Suppose that T contains an edge $e = v_1v_2$ such that neither v_1 nor v_2 is a leaf. Let $a := (v_1, v_2)$. Deletion of e splits T into subtrees T_1 and T_2 with $v_1 \in VT_1$ and $v_2 \in VT_2$. Define S_1 (resp. S_2) to be the set of terminals $s \in S$ such that T_s is entirely contained in VT_1 (resp. in VT_2). Then $S_1 \cap S_2 = \emptyset$ and each terminal in $S - (S_1 \cup S_2)$ is complex (by properties (C1),(C3)). Hence $\mathcal{N}$ is Eulerian at each vertex in $VG - (S_1 \cup S_2)$. Also from the definition of Π_a it follows that

$$\Pi_a = S_1 \times S_2 \quad \text{and} \quad \Pi_{\overline{a}} = S_2 \times S_1.$$

Compute an (S_1, S_2)-cut (X_1, X_2) of minimum capacity $c(\delta^{\mathrm{out}}(X_1))$ in G. Then (X_1, X_2) is Π_a-separating and (X_2, X_1) is $\Pi_{\overline{a}}$-separating. The Eulerianess implies

$$c(\delta^{\mathrm{out}}(X_1)) - c(\delta^{\mathrm{in}}(X_1)) = \sum\nolimits_{s \in S_1} c(\delta^{\mathrm{out}}(s)) - c(\delta^{\mathrm{in}}(s)).$$

Hence the capacity $c(\delta^{\mathrm{out}}(X_2)) = c(\delta^{\mathrm{in}}(X_1))$ is minimum among all (S_2, S_1)-cuts in G as well.

We construct two new instances $(\mathcal{N}_1, \mu_1)$ and $(\mathcal{N}_2, \mu_2)$ in a natural way. More precisely, set $\mathcal{N}_1 := (G_1, S_1', c_1)$, where G_1 is obtained from G by contracting X_2 into a new vertex z_2 (and deleting the loops if appeared), c_1 is the restriction of c to the arc set of G_1, and $S_1' := (S \cap X_1) \cup \{z_2\}$. The distance μ_1 is induced by the tree T_1' obtained from T by contracting the subtree T_2 into v_2; the arc lengths in T_1' are same as in T. (In fact, these lengths are ignored by the algorithm and they are needed only for our analysis.) Terminals $s \in S \cap X_1$ are now realized by the subtrees of T_1' obtained by restricting the subtrees T_s in $\mathcal{R}$ to T_1'. The terminal z_2 is realized by $\{v_2\}$. Let $\mathcal{R}_1$ denote the resulting realization of μ_1.

The construction of $\mathcal{N}_2 = (G_2, S_2', c_2)$, μ_2, $\mathcal{R}_2$ is symmetric (by swapping $1 \leftrightarrow 2$).

The algorithm recursively constructs integer optimal multiflows F_1 and F_2 for $(\mathcal{N}_1, \mathcal{R}_1)$ and $(\mathcal{N}_2, \mathcal{R}_2)$, respectively. The following property easily follows from the minimality of (X_1, X_2) and (X_2, X_1):

(2.6) the multiflow F_1 saturates the cuts $\delta^{\mathrm{in}}_{G_1}(z_2)$ and $\delta^{\mathrm{out}}_{G_1}(z_2)$; similarly, F_2 saturates $\delta^{\mathrm{in}}_{G_2}(z_1)$ and $\delta^{\mathrm{out}}_{G_2}(z_1)$.

This property enables us to "glue" (or "aggregate") F_1 and F_2 into an integer multiflow F in $\mathcal{N}$ which saturates both cuts (X_1, X_2) and (X_2, X_1) (being Π_a-separating and $\Pi_{\overline{a}}$-separating cuts, respectively). These cuts together with the preimages in G of corresponding saturated cuts for F_1 and F_2 give a collection of saturated cuts for F as required in (2.2), yielding the optimality of F by Lemma 2.1.

The above partitioning step reduces the current problem instance to a pair of smaller ones (in particular, the tree sizes strictly decrease). One easily checks that conditions (C1)–(C5) (see Section 2.1) are maintained. Note that for $i \in \{1, 2\}$, if s

is a complex terminal in S such that the image of T_s in T'_i is different from $\{v_{3-i}\}$, then we should keep regarding s as a complex terminal in $\mathcal{N}_i$ (even if this image is a nontrivial (undirected) path having zero ℓ-length in one direction). This is not confusing since the network continues to be Eulerian at s.

The recursion process with a current T stops when each edge in it is incident to a leaf. Since each inner vertex of T has degree 3 (by (C4)), only two cases of T are possible:

(i) VT consists of two vertices v_1 and v_2;

(ii) VT consists of one inner vertex v_0 and three leaves v_1, v_2, v_3.

Case (i) is considered in Subsection 2.4, and case (ii) in Subsection 2.5.

2.4 Basic step: two vertices

Let $e = v_1v_2$ be the only edge of T. Note that the vertices v_1 and v_2 may correspond to many terminals in S. Let terminals $s_1, \dots, s_p$ (resp. $t_1, \dots, t_q$) be realized in $\mathcal{R}$ by $\{v_1\}$ (resp. $\{v_2\}$). Also there may exist a terminals s realized by the whole tree T; but such an s may be ignored since $\mu(s,t) = \mu(t,s) = 0$ for any $t \in S$.

Let $S' := \{s_1, \dots, s_p\}$ and $T' := \{t_1, \dots, t_q\}$. Construct an integer maximum S'–T' flow, i.e. a function $f\colon AG \to \mathbb{Z}_+$ with $\mathrm{val}(f) := \sum(\mathrm{div}_f(s) : s \in S')$ maximum subject to $f(a) \le c(a)$ for each $a \in AG$ and $\mathrm{div}_f(v) = 0$ for each $v \in VG - (S' \cup T')$. Here $\mathrm{div}_f(v)$ denotes the divergence $f(\delta^{\mathrm{out}}(v)) - f(\delta^{\mathrm{in}}(v))$. Then f saturates some (S',T')-cut $(X, \overline{X})$. Since capacities c are Eulerian at all inner vertices, $g := c - f$ is a T'–S' flow. This implies that $\mathrm{val}(g) = c(\overline{X}, X)$ and that g saturates the reversed cut $(\overline{X}, X)$.

We construct F by combining path decompositions of f and g. Let $a := (v_1, v_2)$. Then $\Pi_a = S' \times T'$, and $(X, \overline{X})$ is a Π_a-separating cut. The multiflow $F[\Pi_a]$ corresponds to f, and therefore it saturates $(X, \overline{X})$. Similarly, $\Pi_{\overline{a}} = T' \times S'$, $(\overline{X}, X)$ is a $\Pi_{\overline{a}}$-separating cut, the multiflow $F[\Pi_{\overline{a}}]$ corresponds to g, and therefore it saturates $(\overline{X}, X)$. This gives (2.2) for F.

2.5 Basic step: three leaves

This case is less trivial. Here ET consists of three edges $e_i = v_iv_0$, $i = 1, 2, 3$. We denote the arc (v_i, v_0) by a_i.

Let us call terminals s, s' in the current network $\mathcal{N} = (G, S, c)$ *similar* if they are realized by the same subtree of T; clearly $\mu(s,p) = \mu(s',p)$ and $\mu(p,s) = \mu(p,s')$ for any $p \in S$. Suppose that there are similar *simple* terminals s, s'. They correspond to the singleton $\{v_i\}$ for some $i \in \{1,2,3\}$ (in view of (C3)). The fact that v_i is a leaf of T provides the triangle inequality $\mu(p,s) + \mu(s,q) \ge \mu(p,q)$ for any $p, q \in S$, and similarly for s'. Due to this, we can identify s, s' in G into one terminal (corresponding to $\{v_i\}$) without affecting the problem in essence.

Thus, we may assume that for each $i = 1, 2, 3$, there is exactly one terminal, s_i say, corresponding to $\{v_i\}$. Let $S' := \{s_1, s_2, s_3\}$. Note that each terminal $s \in S - S'$ is (regarded as) complex, and c is Eulerian at s. We partition $S - S'$ into subsets $S_1, S_2, S_3, S_{12}, S_{13}, S_{23}$, where S_i (resp. S_{ij}) consists of the (similar) terminals corresponding to the subtree of T induced by the edge e_i (resp. by the pair $\{e_i, e_j\}$).

Suppose we ignore the terminals in $S - S'$, by considering the network $\mathcal{N}' := (G, S', c)$. This network is inner Eulerian since $\mathcal{N}$ is Eulerian within $S - S'$. Using the algorithm from [BK07], we find an optimal multiflow F to problem **IMF** for $\mathcal{N}'$ with unit distance for each pair (s_i, s_j), $i \neq j$. Also for $i = 1, 2, 3$, we find a minimum capacity $(s_i,\ S' - s_i)$-cut $(X_i, \overline{X}_i)$ in $\mathcal{N}'$. They can be chosen so that the sets X_1, X_2, X_3 are pairwise disjoint. Also one may assume that each path P with $F(P) > 0$ is simple and has no intermediate vertex in S'. Then F yields a solution to $(\mathcal{N}', \mu')$, where μ' is the restriction of μ to $S' \times S'$. Since $\mathcal{N}'$ is inner Eulerian and in view of Theorem 1.1, F saturates both cuts $(X_i, \overline{X}_i)$ and $(\overline{X}_i, X_i)$ for each i. Associating such cuts to the arcs $a_i, \overline{a}_i$ results in (2.2). Then F is optimal by Lemma 2.1.

Next we return to $\mathcal{N}$ as before. The above multiflow F need not be optimal for $(\mathcal{N}, \mu)$ since cuts $(X_i, \overline{X}_i)$ may not be Π_{a_i}-separating for some i. Our aim is to improve F, X_1, X_2, X_3 so as to ensure (2.2).

More precisely, we are looking for subsets $X'_i \subseteq X_i$, $i = 1, 2, 3$, and a multiflow F' such that:

(2.7) (i) $S_i \subset VG - (X_j \cup X_k)$ and $S_{ij} \subset VG - X_k$ for any distinct i, j, k;

(ii) for $i = 1, 2, 3$, the cuts $(X'_i, \overline{X}'_i)$ and $(\overline{X}'_i, X'_i)$ are saturated by F';

(iii) each path P with $F'(P) > 0$ connects either s_i and s_j, or s_i and S_j, or s_i and S_{jk}, where i, j, k are distinct.

By (2.7)(i), the cut $(X'_i, \overline{X}'_i)$ is Π_{a_i}-separating, $i = 1, 2, 3$. In their turn (2.7)(ii),(iii) imply that $F'[\Pi_{a_i}]$ saturates $(X'_i, \overline{X}'_i)$. Then F' is optimal by Lemma 2.1.

We construct the desired X'_i and F' as follows. For $i = 1, 2, 3$, let Q_i denote the set of terminals s that violate (2.7)(i) w.r.t. X_i, i.e. $s \in X_i$ but $s \notin \{s_i\} \cup S_i \cup S_{ij} \cup S_{ik}$ (where $\{i, j, k\} = \{1, 2, 3\}$). If $Q_i = \emptyset$ then $(X_i, \overline{X}_i)$ is already Π_{a_i}-separating, in which case we set $X'_i := X_i$.

Let $Q_i \neq \emptyset$. We construct the digraph G_i from G by contracting $VG - X_i$ into a new terminal z_i. Arc capacities in G_i are induced by those in G (and are denoted by c as before). This gives the network $\mathcal{N}_i := (G_i, \{s_i, z_i\} \cup Q_i, c)$ which is Eulerian at all vertices except, possibly, for s_i and z_i. The current multiflow F in G induces a multiflow F_i in G_i consisting of weighted s_i–z_i and z_i–s_i paths. Since F saturates $\delta^{\rm in}_G(X_i)$ and $\delta^{\rm out}_G(X_i)$, the multiflow F_i saturates $\delta^{\rm in}_{G_i}(z_i)$ and $\delta^{\rm out}_{G_i}(z_i)$.

Now we find in G_i a maximum integer flow g_i from the source s_i to the set of sinks $Q_i \cup z_i$. Moreover, among such flows we choose one maximizing $-\operatorname{div}_{g_i}(z_i)$. (This is done by standard flow techniques: take a maximum s_i–z_i flow (e.g. by extracting the subflow in F_i formed by s_i–z_i paths), then switch to the residual network and augment the current flow to get a maximum s_i–$(Q_i \cup z_i)$ flow.) The flow g_i is decomposed into a collection of weighted s_i–z_i paths, denoted by $\widehat{g}_i(s_i, z_i)$, a collection of weighted s_i–t paths for $t \in Q_i$, denoted by $\widehat{g}_i(s_i, t)$, ignoring possible cycles. By the construction, g_i saturates the trivial cut $\delta^{\text{in}}_{G_i}(z_i)$ and some $(s_i, Q_i \cup z_i)$-cut $\delta^{\text{out}}_{G_i}(X'_i)$.

It remains to construct flows on paths going in the opposite direction, i.e. entering s_i. Define the function $h_i := c - g_i$ on AG_i. It is Eulerian at all vertices in $VG_i - (\{s_i, z_i\} \cup Q_i)$. Also $\text{div}_{h_i}(z_i) \geq 0$ (since g_i saturates $\delta^{\text{in}}_{G_i}(z_i)$) and $\text{div}_{h_i}(t) \geq 0$ for all $t \in Q_i$ (since c is Eulerian at t). We decompose h_i into a collection of weighted z_i–s_i paths, denoted by $\widehat{h}_i(z_i, s_i)$), and a collection of weighted t–s_i paths for $t \in Q_i$, denoted by $\widehat{h}_i(t, s_i)$, ignoring possible cycles. These paths saturate $\delta^{\text{in}}_{G_i}(X'_i)$ (since $h_i(\delta^{\text{out}}_{G_i}(X'_i)) = 0$) and $\delta^{\text{out}}_{G_i}(z_i)$ (since $h_i(\delta^{\text{in}}_{G_i}(z)) = 0$).

The collections $\widehat{g}_i(\cdot)$ and $\widehat{h}_i(\cdot)$ constitute a multiflow F'_i that replaces the "restriction" F_i of F on G_i. Making such "replacements" for $i = 1, 2, 3$ (and using the fact that X_1, X_2, X_3 are disjoint), we obtain an integer multiflow F' which along with X'_1, X'_2, X'_3 as above satisfies (2.7). Hence F' is optimal.

2.6 Complexity

In this final section we describe an efficient implementation of our algorithm and estimate its complexity. Current multiflows in the process are stored as collections of point-to-point flows. Namely, an integer multiflow F in a network with terminals S is maintained as a collection $\{f_{st} \mid s, t \in S, s \neq t\}$, where f_{st} is an integer s–t flow (called an *s–t component* of F).

Let $\varphi(n, m, k)$ denote the complexity of the algorithm applied to an instance with n vertices and m arcs of G, and k leaves of T.

The case $k = 2$ was studied in Subsection 2.4. The algorithm involves a single max-flow computation and two flow decompositions. Hence

$$\varphi(n, m, 2) = O(MF(n, m) + mn), \tag{2.8}$$

where $MF(n', m')$ denotes the complexity of a max-flow algorithm in a network with n' vertices and m' arcs.

The case $k = 3$ was considered in Subsection 2.5. It reduces to solving a three-terminal version of the unweighted directed **IMF** problem followed by $O(1)$ max-flow computations and decompositions. With the help of the algorithm from [BK07] the three-terminal multiflow problem is solved in $O(MF(n, m) + mn \log(n^2/m))$ time. Therefore,

$$\varphi(n, m, 3) = O(MF(n, m) + mn \log(n^2/m)). \tag{2.9}$$

For $k \geq 4$, we apply the partitioning operation from Subsection 2.3. Computing a minimum cut dividing the current instance $\mathcal{N}$ into $\mathcal{N}_1$ and $\mathcal{N}_2$ takes $O(MF(n, m))$ time. The aggregation takes the s–z_2 components of F_1 (for $s \in S \cap X_1$) and the z_1–s components of F_2 (for $s \in S \cap X_2$), combines them into an $(S \cap X_1)$–$(S \cap X_2)$ flow and decomposes it into a collection of flows for all source-sink pairs. The algorithm similarly handles the z_2–s components of F_1 (for $s \in S \cap X_1$) and the s–z_1 components of F_2 (for $s \in S \cap X_2$). Finally it adds remaining components of F_1 and F_2, thus forming an integer optimal multiflow in $\mathcal{N}$. In total the aggregation operations take $O(mn)$ time, hence

$$\varphi(n,m,k) = \varphi(n_1,m_1,k_1) + \varphi(n_2,m_2,k_2) + O(MF(n,m) + mn), \tag{2.10}$$

where (n_i, m_i, k_i) are the size parameters for $\mathcal{N}_i$.

Since degrees of inner nodes of T are 3 by (C3), there exists (and can be found in $O(k)$ time) a partitioning edge in T that yields $k_1, k_2 \le 2k/3 + 1$. Thus, the height of the recursion tree is at most $O(\log S)$. Also $n_1 + n_2 = n + 2$ and $m_1, m_2 \le m$. Assuming that $MF(n,m)$ obeys some technical conditions (e.g., satisfying $MF(n,m) = O(mn \log(n^2/m))$ as in the algorithm of Goldberg and Tarjan [GT88]), one can show by induction that (2.8), (2.9), and (2.10) imply

$$\varphi(n,m,k) = O((MF(n,m) + mn)\log k + mn \log(n^2/m))$$

(applying reasonings similar to those in [IKN98]). By spending additional $O(mn \log |S|)$ time, one can convert the resulting integer optimal multiflow into path-packing form, as exlained in [BK07]. In total, the algorithm takes $O((MF(n,m) + mn)\log |S| + mn \log(n^2/m))$ time, as declared.

References

[FHW80] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Sci.*, 10:111–121, 1980.

[BK07] M.A. Babenko and A.V. Karzanov. Free multiflows in bidirected and skew-symmetric graphs. *Discrete Appl. Math.*, 155(13):1715–1730, 2007.

[Fr89] A. Frank. On connectivity properties of Eulerian digraphs. *Ann. Discrete Math.*, 41:179–194, 1989.

[GT88] A.V. Goldberg and R.E. Tarjan, A new approach to the maximum flow problem, *J. ACM*, 35:921–940, 1988.

[IKN98] T. Ibaraki, A.V. Karzanov, and H. Nagamochi. A fast algorithm for finding a maximum free multiflow in an inner Eulerian network and some generalizations. *Combinatorica*, 18(1):61–83, 1998.

[HK10] H. Hirai and S. Koichi. On duality and fractionality of multicommodity flows in directed networks. *ArXiv*:1006.5520[math.CO], 2010.