COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **18** (4) (1998) 549–568

A COMBINATORIAL ALGORITHM FOR THE MINIMUM $(2,r)$-METRIC PROBLEM AND SOME GENERALIZATIONS*

ALEXANDER V. KARZANOV

Received June 11, 1998

Let $G=(V,E)$ be a graph with nonnegative integer capacities $c(e)$ of the edges $e\in E$, and let μ be a metric that establishes distances on the pairs of elements of a subset $T\subseteq V$. In the *minimum 0-extension problem* $(*)$, one is asked for finding a (semi)metric m on V such that m coincides with μ within T, each $x\in V$ is at zero distance from some $t\in T$, and the value $\sum(c(e)m(e):e\in E)$ is as small as possible. This is the classical minimum (undirected) cut problem when $T=\{s,t\}$ and $\mu(s,t)=1$, and the *minimum* $(2,r)$*-metric problem* when μ is the path metric of the complete bipartite graph $K_{2,r}$. It is known that the latter problem can be solved in strongly polynomial time by use of the ellipsoid method.

We develop a polynomial time algorithm for the minimum $(2,r)$-metric problem, using only "purely combinatorial" means. The algorithm simultaneously solves a certain associated integer multiflow problem. We then apply this algorithm to solve $(*)$ for a wider class of metrics μ, present other results and raise open questions.

1. Introduction

By a *metric* on a set S we mean a function $d:S\times S\to\mathbb{R}_+$ that establishes *distances* on the pairs of elements of S satisfying (i) $d(x,x)=0$, (ii) $d(x,y)=d(y,x)$, and (iii) $d(x,y)+d(y,z)\geq d(x,z)$, for all $x,y,z\in S$. We usually write $d(xy)$ in place of $d(x,y)$ and allow $d(xy)=0$ for some $x\neq y$. A special case of metrics is the *path metric* d^Γ of a connected graph $\Gamma=(S,W)$, i.e., $\mathrm{d}^\Gamma(xy)$ is the minimum number of edges of a path in Γ connecting nodes x and y.

We consider an undirected graph $G=(V,E)$ whose edges $e\in E$ have nonnegative integer *capacities* $c(e)$, and a subset $T\subseteq V$ of nodes called *terminals*. Let μ be a metric on T. A metric m on V is said to be an *extension* of μ to V if the submetric of m on T is just μ, and to be a *0-extension* if, in addition, for each $x\in V$, there is $t\in T$ such that $m(tx)=0$. If μ is a *positive* metric, i.e., $\mu(st)>0$ for any $s\neq t$, then each 0-extension m of μ to V one-to-one corresponds to a T-*partition*, i.e., a partition of V into $|T|$ subsets X_t, $t\in T$, such that each X_t contains exactly one terminal, namely, t. This correspondence is defined by $X_t=\{x\in V:m(tx)=0\}$.

Mathematics Subject Classification (1991): 90C27; 90B10

* This research was supported by grant 97-01-00115 from the Russian Foundation of Basic Researches and a grant from the Sonderforschungsbereich 343, Bielefeld Universität, Bielefeld.

A path in G whose ends are different elements of T is called a T-*path*. A (c-admissible) *multiflow* f consists of T-paths $P_1,\dots,P_k$ in G along with nonnegative real weights $\lambda_1=\lambda(P_1),\dots,\lambda_k=\lambda(P_k)$ satisfying the capacity constraint

$$f^e := \sum(\lambda_i\ :\ P_i \text{ contains } e) \le c(e) \text{ for each } e\in E.$$

Define $\langle\mu,f\rangle$, the μ-*value* of f, to be $\sum(\mu(u_iv_i)\lambda_i : i=1,\dots,k,\ u_i$ and v_i are the ends of P_i). If all λ_i's are integers, f is called an *integer* multiflow. Consider the following four problems:

(1.1) Find a 0-extension m of μ to V with $c\cdot m:=\sum(c(e)m(e):e\in E)$ minimum;

(1.2) Find an extension m of μ to V with $c\cdot m$ minimum;

(1.3) Find an integer multiflow f whose μ-value is maximum;

(1.4) Find a multiflow f whose μ-value is maximum.

(Problem (1.1) is also known as the *multifacility location problem*, cf. [17].) Let $\tau=\tau(G,c,\mu)$ and $\tau^*=\tau^*(G,c,\mu)$ denote the minimum $c\cdot m$ in (1.1) and (1.2), respectively, and let $\nu=\nu(G,c,\mu)$ and $\nu^*=\nu^*(G,c,\mu)$ denote the maximum μ-values in (1.3) and (1.4), respectively. An extension m with $c\cdot m=\tau^*$ is called *minimum*, and a multiflow f with $\langle\mu,f\rangle=\nu^*$ is called *maximum*, and similarly for 0-extensions and integer multiflows. Since (1.1) is a strengthening of (1.2), and (1.3) is a strengthening of (1.4), we have $\tau\ge\tau^*$ and $\nu\le\nu^*$. In their turn, the relaxations (1.2) and (1.4) are, in fact, mutually dual linear programs (see, e.g., [12]), whence $\tau^*=\nu^*$. Thus, we have the following relations:

$$\tau\ge\tau^*=\nu^*\ge\nu. \tag{1.5}$$

Each of the two inequalities here may be strict. The simplest case with equality throughout arises when $\mu=\mathrm{d}^{K_2}$ (K_p is the complete graph with p nodes). In this case any 0-extension is a cut metric, an optimal 0-extension corresponds to a minimum cut "separating" the pair of terminals, and $\tau=\nu$ holds by the classical max flow min cut theorem [7]. On the other hand, if $\mu=\mathrm{d}^{K_p}$ with $p\ge 3$, then τ may differ from τ^*; in this case (1.1) turns into the *minimum multiterminal cut problem*, which is known to be strongly NP-hard even if $p=3$ [5].

Following [13], a metric μ on T is called *minimizable* if $\tau(G,c,\mu)=\tau^*(G,c,\mu)$ holds for any graph $G=(V,E)$ with $V\supseteq T$ and capacities $c:E\to\mathbb{Z}_+$. For such a μ, problem (1.1) can be solved in strongly polynomial time by use of the ellipsoid method, taking into account that (1.2) is a linear program whose constraint matrix size is polynomial in $|V|,|E|$. The class of minimizable metrics is rather large; in particular, it includes $\mu=\mathrm{d}^{K_{2,r}}$ for any r, where $K_{p,q}$ is the complete bipartite graph whose parts (the maximal stable sets) consist of p and q nodes. For such a μ, a 0-extension is called a $(2,r)$-*metric*, and (1.1) can be specified as:

(1.6) Given G,T,c and a partition of T into two subsets $A=\{s_1,s_2\}$ and $B=\{t_1,\dots,t_r\}$, find a T-partition $\{S_1,S_2,T_1\dots,T_r\}$ of V with $s_i\in S_i$ and $t_j\in T_j$ that minimizes $\sum(c(S_i,T_j):i=1,2,\,j=1,\dots,r)+2c(S_1,S_2)+2\sum(c(T_i,T_j):1\le i<j\le r)$,

where for $X,Y \subseteq V$, $c(X,Y)$ denotes the total capacity of edges with one end in X and the other in Y. Although the second inequality in (1.5) may be strict for $\mu = \mathrm{d}^{K_{2,r}}$ too, it holds with equality in the following important case. We say that c is *inner Eulerian* if $c(X, V-X)$ is an even integer for any $X \subseteq V-T$.

Theorem 1.1 [13]. *If c is inner Eulerian and $\mu = \mathrm{d}^{K_{2,r}}$, then $\nu = \nu^*$ and, therefore, $\nu = \tau$.*

The main aim of this paper is to give a "purely combinatorial" algorithm which finds a minimum $(2,r)$-metric and finds a maximum integer multiflow when c is inner Eulerian. The algorithm we develop runs in time polynomial in $|V|,|E|$ and linear in $\log ||c||$, where $||c|| = \max\{c(e) : e \in E\}$.

In fact, the algorithm focuses on construction of a maximum integer multiflow, whereas a minimum $(2,r)$-metric is obtained as a by-product. It involves three ingredients: (i) a capacity scaling method, (ii) an integer augmentation procedure, and (iii) a maximality check-up procedure. These occur in the high, middle and low levels of the algorithm, respectively. The capacity scaling method reduces the whole problem to about $\log ||c||$ similar problems each of which deals with a truncated capacity function c' and finds a maximum integer multiflow for it, starting with a nearly optimal integer multiflow f' whose μ-value is only $O(|E|)$ below $\tau\langle G, c', \mu\rangle$. Therefore, at most $O(|E|)$ integer augmentations of the μ-value are sufficient to transform f' into a maximum integer multiflow for c'. The integer augmentation procedure is somewhat more complicated than that based on standard augmenting path techniques in maximum flow algorithms. It involves a vertex splitting method and relies on possibility to decide whether a given (fractional) multiflow for a given capacity function is maximum or not. Our maximality check-up procedure solves the latter problem in strongly polynomial time and in a combinatorial fashion.

Next we consider a more general case. A complete characterization of the class of graphs whose path metrics are minimizable is exhibited in the following theorem.

Theorem 1.2 [11]. *For a graph H, the metric d^H is minimizable if and only if H is bipartite, orientable and contains no isometric k-circuit with $k \geq 6$.*

Here a subgraph (or circuit) $H' = (T', U')$ of H is called *isometric* if $\mathrm{d}^{H'}(uv) = \mathrm{d}^H(uv)$ for all $u,v \in T'$; a *k-circuit* is a (simple) circuit with k nodes; and H is called *orientable* if its edges can be oriented so that for any 4-circuit $C = (v_0, e_1, v_1, \ldots, e_4, v_4 = v_0)$, the orientations of the opposite edges e_1 and e_3 are different along the circuit, and similarly for e_2 and e_4; a feasible orientation for C is shown in Fig. 1a. For example, the graph $K_{p,r}$ is orientable if and only if $\min\{p,r\} \leq 2$. A graph H as in Theorem 1.2 is called a *frame*; we usually assume, w.l.o.g., that H has no parallel edges or loops.

We say that edges e, e' of a graph $H = (T,U)$ are *dependent* if there is a sequence $e = e_0, e_1, \ldots, e_k = e'$ where each two consecutive e_i, e_{i+1} are opposite edges in some 4-circuit of H. This relation is symmetric and transitive, and a maximal set of dependent edges is called an *orbit* of H. For example, $K_{2,r}$ has two orbits when

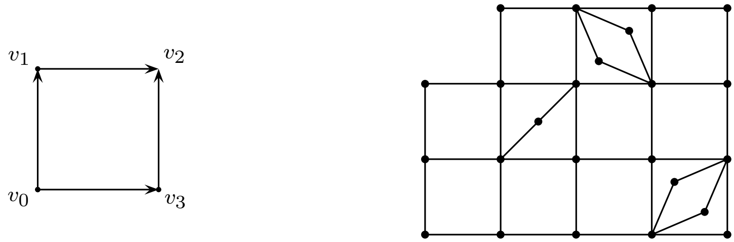

Fig. 1. (a) (b)

$r=2$ and one orbit when $r\geq 3$. For $Z\subseteq U$, let H/Z denote the graph obtained from H by contracting each edge in Z.

Definition 1.3. A frame $H=(T,U)$ with the orbits $Q_1,\dots,Q_k$ is called *sparse* if for each i, the graph H_i obtained from $H/(U-Q_i)$ by identifying the parallel edges and deleting the loops (if any) is $K_{p,r}$ with $\min\{p,r\}\leq 2$.

Remark. One can show that a frame is sparse if and only if each orbit contains (the edge set of) at most one maximal subgraph $K_{2,r}$ with $r\geq 3$.

Figure 1b illustrates a sparse frame with four orbits. We show that for any sparse frame H, (1.1) with $\mu=\mathrm{d}^H$ is reduced to $O(|T|)$ minimum cut or minimum $(2,r)$-metric computations. This reduction and our minimum $(2,r)$-metric algorithm provide a "purely combinatorial" polynomial time algorithm for such a μ.

This paper is organized as follows. The algorithm for the minimum $(2,r)$-metric problem and its corresponding integer multiflow problem in the inner Eulerian case is described throughout Sections 2–4. Section 5 is devoted to the generalization for sparse frames. The concluding Section 6 contains some additional result and raises open questions.

Throughout, for $X\subseteq V$, $\delta(X)=\delta^G(X)$ denotes the set of edges of G with one end in X and the other in $V-X$ (a *cut* in G), and $c(\delta(X))$ stands for $c(X,V-X)$. An unordered pair $\{u,v\}$ of nodes is usually abbreviated as uv. For a multiflow f in a network (G,c,T) and a pair uv in T, f_{uv} denotes the part of f formed by the paths with the ends u and v (the *flow* between u and v); the total weight of these paths is called the *value* of f_{uv} and denoted by $|f_{uv}|$. Hence, the μ-value of f is $\sum(\mu(uv)|f_{uv}|:\{u,v\}\subseteq T)$. For a path P in G, χ^P denotes its incidence vector in $\mathbb{R}^E$, i.e., $\chi^P(e)$ is the number of occurrencies of an edge $e\in E$ in P.

In conclusion of this section we recall one application of the minimum $(2,r)$-metric problem. In the *multiflow demand problem*, one is given a set D of pairs uv of terminals and *demands* $d(uv)\in\mathbb{Z}_+$ on these pairs, and is asked to find a multiflow f in (G,c,T) satisfying $|f_{uv}|=d(uv)$ for all $uv\in D$. By a five-terminus flow theorem [9], when $|T|=5$, the demand problem has a solution if and only if the cut condition

$$c(\delta(X))\geq\sum(d(uv)\ :\ uv\in D,\ |\{u,v\}\cap X|=1)=:d_X \tag{1.7}$$

holds for each $X \subset V$, and the (2,3)-metric condition

$$c \cdot m \geq \sum (d(uv)m(uv) \,:\, uv \in D) \tag{1.8}$$

holds for each (2,3)-metric m on V (concerning all possible graphs $H \simeq K_{2,3}$ on T). Also if (c,d) satisfies the partity condition

$$c(\delta(X)) + d_X \equiv 0 \pmod 2 \qquad \text{for all } X \subset V, \tag{1.9}$$

then the problem has an integer solution f provided that it has a solution at all; moreover, to find such an f takes $O(|V|^3)$ minimum cut and minimum (2,3)-metric computations. Therefore, f can be found in polynomial time by a "purely combinatorial" algorithm.

2. Checking the maximality

Let $f = (P_1, \ldots, P_k, \lambda_1, \ldots, \lambda_k)$ be a multiflow for G, T, c, and let $\mu = \mathrm{d}^H$, where $H = (T, U)$ is $K_{2,r}$. We assume that $\lambda_1, \ldots, \lambda_k > 0$. The algorithm described in this section decides whether the given f is maximum, and if so, finds a minimum $(2,r)$-metric. It is based on a "flow expansion" idea borrowed from [14].

The algorithm transforms f step by step, and it may happen that some paths of the current multiflow are self-intersecting in edges even if all paths in the initial multiflow are simple. By this reason, we should refine the definition of f^e in the Introduction as

$$f^e := \sum (\lambda_i n_i(e) \,:\, i = 1, \ldots, k), \tag{2.1}$$

where $n_i(e)$ is the number of occurrencies of e in P_i. We define $\Delta(e) = \Delta_f(e) := c(e) - f^e$ and call e *residual* if it is not saturated by f, i.e., if $\Delta(e) > 0$. Let $E^0 = E^0(f)$ be the set of residual edges in G. A *residual path* is a path in G whose all edges are residual. When it is not confusing, a path $P = (x_0, e_1, x_1, \ldots, e_k, x_k)$ is abbreviated as $x_0 x_1 \ldots x_k$; we say that P is an x_0–x_k *path*. Each path in f is usually considered up to reversing. The complete graph on T is denoted by $K_T = (T, E_T)$. For $uv \in E_T$, the set of nodes occurring in paths of the flow f_{uv} is denoted by V_{uv} (the *domain* of f_{uv}). Let $A = \{s_1, s_2\}$ and $B = \{t_1, \ldots, t_r\}$ be the parts of H.

Each transformation of f never decreases the μ-value or any of the sets E^0 and V_{uv}, $uv \in E_T$; moreover, it increases at least one of these. It applies one of the seven operations below. The algorithm declares that the initial multiflow is not maximum and terminates when the μ-value becomes larger (the *breakthrough* situation). This happens after performing the X-operation and, sometimes, the V-operation below. Also the breakthrough happens if there appears a residual path P connecting different terminals (for we can increase the μ-value by pushing some nonzero flow through P).

I-operation. Suppose there are a residual edge xy and a flow f_{uv} with V_{uv} containing x but not y. The *I-operation* chooses a path $P=x_0\dots x_k$ in f_{uv} passing x, $x=x_i$ say, adds to f_{uv} the path $P'=x_0\dots x_i y x_i\dots x_k$ taken with a sufficiently small weight $\varepsilon>0$ satisfying $\varepsilon<\lambda(P),\Delta(e)/2$, and accordingly reduces the weight of P by ε. This adds y to V_{uv}, while preserving the c-admissibility, the μ-value, and the sets E^0 and $V_{u'v'}$ for all other $u'v'\in E_T$.

V-operation. Suppose there are a u–v path $P=x_0\dots x_k$ in f and a (simple) residual path $L=z_0\dots z_d$ such that: $z_0=z$ is a terminal different from u and v; $z_d=x_i$ for some i; and either (a) $\mu(uz)+\mu(zv)>\mu(uv)$, or (b) some node x of the path $P'=x_0\dots x_i z_{d-1}\dots z_0$ is not in V_{uz}, or both. The *V-operation* adds to f the u–z path P' and the z–v path $P''=z_0\dots z_d x_{i+1}\dots x_k$, each taken with a weight $\varepsilon>0$ such that $\varepsilon<\lambda(P)$ and $\varepsilon<\Delta(e)/2$ for all edges e of L, and reduces $\lambda(P)$ by ε. This increases the μ-value, yielding the breakthrough, in case (a), and increases V_{uz} in case (b) (since x is added to V_{uz}).

O-operation. Suppose there are a u–v path $P=x_0\dots x_k$ in f and a residual path $L=z_0\dots z_d$ such that $z_0=x_i$ and $z_d=x_j$ for some $0\le i<j\le k$, and some edge e of the subpath $x_i\dots x_j$ is saturated. The *O-operation* adds to f the u–v path $x_0\dots x_i z_1\dots z_{d-1}x_j\dots x_k$ with a weight $\varepsilon>0$ such that $\varepsilon<\lambda(P),\Delta(z_{p-1}z_p)$, $p=1,\dots,d$, and reduces $\lambda(P)$ by ε. This increases E^0 (since e becomes residual).

Y-operation. Suppose there are a u–v path $P=x_0\dots x_k$ and a u–v' path $Q=y_0\dots y_q$ in f with $v\ne v'$ such that $x_i=y_j$ for some $i,j>0$, and some node of the subpath $y_0\dots y_j$ is not in V_{uv}. The *Y-operation* adds to f the paths $y_0\dots y_j x_{i+1}\dots x_k$ and $x_0\dots x_i y_{j+1}\dots y_q$, each taken with a weight ε, $0<\varepsilon<\lambda(P),\lambda(Q)$, and reduces each of $\lambda(P)$ and $\lambda(Q)$ by ε. This increases V_{uv}.

U-operation. Suppose there are a u–v path $P=x_0\dots x_k$ and a u–v' path $Q=y_0\dots y_q$ in f such that: $x_i=y_j$ for some $i,j>0$; $\mu(uv)+\mu(uv')=\mu(vv')$; and some edge e of the closed path $x_0\dots x_i y_{j-1}\dots y_0$ is saturated. The *U-operation* adds to f the v–v' path $L=x_k x_{k-1}\dots x_i y_{j+1}\dots y_q$ with a weight ε, $0<\varepsilon<\lambda(P),\lambda(Q)$, and reduce each of $\lambda(P)$ and $\lambda(Q)$ by ε. This increases E^0 (since e becomes residual), while preserving the μ-value.

Ψ-*operation.* Suppose there are a u–v path $P=x_0\dots x_k$ and a z–w path $Q=y_0\dots y_q$ in f such that: $x_i=y_j$ for some i,j; the terminals v,z,w are different elements of B; u is in A; and the subpath $x_0\dots x_i$ contains a saturated edge e. Then $\mu(uv)=1$ and $\mu(vz)=\mu(vw)=\mu(zw)=2$. The Ψ-*operation* adds to f the v–z path $L=x_k\dots x_i y_{j-1}\dots y_0$ and the v–w path $L'=x_k\dots x_i y_{j+1}\dots y_q$, each with a weight ε, $0<\varepsilon<\lambda(Q),\frac{1}{2}\lambda(P)$, reduces $\lambda(P)$ by 2ε and reduces $\lambda(Q)$ by ε. This increases E^0 (since e becomes residual), while preserving the μ-value.

X-operation. Suppose there are a u–v path $P=x_0\dots x_k$ and a z–w path $Q=y_0\dots y_q$ in f such that: $x_i=y_j$ for some i,j; u and z are different terminals in A; and v and w are different terminals in B. The *X-operation* adds to f the u–z path $x_0\dots x_i y_{j-1}\dots y_0$ and the v–w path $x_k\dots x_i y_{j+1}\dots y_q$, each with a weight ε,

$0<\varepsilon\le\lambda(P),\lambda(Q)$, and reduces each of $\lambda(P),\lambda(Q)$ by ε. This increases the μ-value by 2ε (since $\mu(uv)=\mu(zw)=1$ and $\mu(uz)=\mu(vw)=2$), yielding the breakthrough.

The process terminates when the above operations are no longer applicable to the current multiflow. We show the following.

Lemma 2.1. *If neither the breakthrough happens nor any of the above operations is applicable, then the current multiflow f (as well as the initial one) is maximum.*

Proof. We first assume that $f_{s_1t_p}$ is nonempty for each $p=1,\dots,r$. Let S_1 be the node set of the component of (V,E^0) containing s_1. For $p=1,\dots r$, define $V_p=V_{s_1t_p}$ and $T_p=V_p-S_1$. Then $S_1\cap T=\{s_1\}$ and $V_p\cap T=\{s_1,t_p\}$; for if S_1 contains a terminal $u\ne s_1$ or V_p contains a terminal $v\ne s_1,t_p$, we have the breakthrough (taking into account that $\mu(s_1t_p)=1<\mu(s_1v)+\mu(vt_p)$, whence the V-operation is applicable). Also $T_p\cap T_{p'}=\emptyset$ for $p\ne p'$ (otherwise we can apply the U-operation to an s_1–t_p path and an s_1–$t_{p'}$ path which share a common node in $T_p\cap T_{p'}$, increasing E^0). Therefore, the sets $S_1,T_1,\dots,T_r$ and $S_2=V-(S_1\cup T_1\cup\ldots\cup T_r)$ form a T-partition π of V; let m be the $(2,r)$-metric on V induced by π. We assert that

(2.2) for $e\in E$, $m(e)>0$ implies $\Delta(e)=0$,

and

(2.3) each z–w path $Q=y_0\dots y_q$ in f is m-shortest, i.e., $m(Q)=\mu(zw)$.

Property (2.2) follows from the fact that all edges of the cut $\delta(S_1)$ are saturated, and similarly for the cuts $\delta(T_p)$ (for if e is a residual edge in some $\delta(T_p)$, then e belongs to $\delta(V_p)$, and we can apply the I-operation, increasing V_p). To see (2.3), consider possible cases (up to reversing Q). Note that $S_1\subset V_p$ for each p (since the I-operation is impossible).

(a) Let $z=s_1$ and $w=t_p$. Then all nodes of Q are in V_p. Also Q meets $\delta(S_1)$ exactly once. Indeed, if $y_i\in S_1$ for some i, then the nodes $y_0,y_1,\dots y_i$ are contained in S_1 (otherwise E^0 can be increased by use of the O-operation). Hence, $m(Q)=1=\mu(s_1t_p)$.

(b) Let both z,w be in B, $z=t_1$ and $w=t_2$ say. Let y_i be the last node in T_1, and y_j the first node in T_2. Then $y_0,\dots,y_i\in T_1$ and $y_j,\dots,y_q\in T_2$. Indeed, suppose $y_{i'}$ is not in T_1 for some $i'<i$. If $y_{i'}$ is in S_1, then the V-operation applied to Q and a residual path from s_1 to $y_{i'}$ increases V_2 by y_i (as $y_i\in T_1$ implies $y_i\notin V_2$). And if $y_{i'}\notin S_1$, then $y_{i'}\notin V_1$ and, therefore, the Y-operation applied to Q and a t_1–s_1 path in f that passes y_i increases V_1 by $y_{i'}$. Now consider the nodes $y_{i+1},\dots,y_{j-1}$. None of them is in T_p for $p\ne 1,2$ (otherwise the Ψ-operation for Q and an s_1–t_p path in f containing such a node increases E_0). If these nodes are entirely contained in one of S_1 and S_2, then, assuming $j>i+1$, we have $m(Q)=m(y_iy_{i+1})+m(y_{j-1}y_j)=1+1=2=\mu(t_1t_2)$, as required (in case $j=i+1$ we also have $m(Q)=2$). Suppose that $y_g\in S_1$ and $y_h\in S_2$ for some $i<g<h<j$ (the case $i<h<g<j$ is symmetric). Let P be a t_2–s_1 path in f which contains y_g. Then the Y-operation for P and Q increases V_2 by y_h.

(c) Let $z=s_2$ and $w=t_p$. Then Q does not meet $V_{p'}$ for $p'\neq p$ (otherwise we can apply the X-operation, increasing the μ-value). So all nodes of Q are in $T_p\cup S_2$. Also $y_i\in S_2$ implies $y_0,\dots,y_i\in S_2$ (otherwise one can apply the Y-operation, increasing V_p). Hence, Q meets exactly one edge between S_2 and T_p, yielding $m(Q)=1=\mu(zw)$.

(d) Let $z=s_1$ and $w=s_2$. Take the last node y_i of Q not in S_2. Then $y_i\in V_p$ for some p, whence $y_0,\dots,y_i\in V_p$ (otherwise one can increase V_p by use of the Y-operation). Next take the last node y_j of Q in S_1. Then $y_0,\dots,y_j\in S_1$ (otherwise one can increase E^0 by use of the O-operation). Now $i=j$ gives $m(Q)=m(y_iy_{i+1})=2$, and $i>j$ gives $m(Q)=m(y_jy_{j+1})+m(y_iy_{i+1})=1+1=2$. Thus, $m(Q)=\mu(zw)$.

In a general case, for each p with $f_{s_1t_p}$ empty, we define T_p to be the node set of the component of (V,E^0) containing t_p. Again one can see that $\{S_1,S_2,T_1,\dots,T_r\}$ is a T-partition of V. We leave it to the reader, as an exercise, to check that (2.2)–(2.3) hold in this case too (by arguing as above or, sometimes, even simpler).

Now (2.2) and (2.3), being the complementary slackness conditions for (1.2) and (1.4), imply $\langle\mu,f\rangle=c\cdot m$. Indeed, let f consist of u_i–v_i paths Q_i, $i=1,\dots q$, whose weights $\lambda_1,\dots,\lambda_q$ are nonzero. Then

$$\begin{aligned}\langle\mu,f\rangle &= \sum_{i=1}^{q}\lambda_i\mu(u_iv_i)\le\sum_{i=1}^{q}\lambda_i m(Q_i)\\ &=\sum_{e\in E}f^e m(e)\le\sum_{e\in E}c(e)m(e)=c\cdot m,\end{aligned}$$

and equality holds throughout in this expression because of (2.2) and (2.3).

Thus, f is a maximum multiflow and m is a minimum $(2,r)$-metric. ∎

Since each operation applied increases the μ-value or E^0 or V_{uv} for some $uv\in E_T$, the number of iterations in the above process is at most $|E|+|V||E_T|$. Let k be the number of paths of the initial multiflow f. W.l.o.g. one may assume that the length (number of edges) of each of these paths is $O(|V|)$. Since each operation creates at most two new paths, the number of paths of any current multiflow is polynomial in $|V|,|E|,k$. Moreover, after each iteration we can rearrange paths of the current multiflow f in an obvious way in order to maintain the length of each path in f to be $O(|V|)$, without decreasing the μ-value and the sets E^0 and V_{uv}, $uv\in E_T$. Then each of the above operations can be performed in time polynomial in $|V|,|E|,k$, whence the running time of the maximality check-up algorithm is polynomial in $|V|,|E|,k$ as well.

Remark. An analysis of the proof of Lemma 2.1 shows that it suffices to apply only those operations that increase the μ-value or the set E^0 or sets $V_{s_1t_j}$ (rather than $V_{s_1s_2}$, $V_{s_2t_i}$ or $V_{t_it_j}$). Such a modification is more efficient because the number of iterations is lowered from $O(|E|+|T|^2|V|)$ to $O(|E|+|T||V|)$.

Another observation is that the weights of paths as well as the values of residual capacities are, in fact, not essential. The algorithm can maintain only the set of paths of the current multiflow and the set E^0 of residual edges, and perform each operation in terms of these sets. In particular, this implies the following important fact (it will be used later).

Corollary 2.2. *Let $f=(P_1,\dots,P_k\,;\lambda_1,\dots,\lambda_k)$ be a multiflow for G,c,T, and let E^0 be the set of residual edges for c and f. Let f' be the multiflow consisting of the same paths $P_1,\dots,P_k$ but taken with weight one each, and let $c'=\chi^{P_1}+\ldots+\chi^{P_k}+\chi^{E_0}$. Then f is maximum for G,c,μ if and only if f' is maximum for G,c',μ.* ∎

3. Integer augmentation

In this section we deal with a special case when $G=(V,E)$ is an *inner Eulerian graph* (i.e., all inner nodes $x\in V-T$ have even degrees) and all capacities $c(e)$ of edges $e\in E$ are ones. We allow G to have multiple edges but not loops and denote $\nu(G,c,\mu)$ by $\nu(G,\mu)$. As before, $T=\{s_1,s_2,t_1,\dots,t_r\}$ and $\mu=\mathrm{d}^H$, where $H=(T,U)$ is $K_{2,r}$ with the parts $A=\{s_1,s_2\}$ and $B=\{t_1,\dots,t_r\}$. In the input of the integer augmentation problem, we are given an *integer* multiflow f whose μ-value is not maximum, and the goal is to augment its μ-value by at least one, i.e., to find an integer multiflow f' with $\langle\mu,f'\rangle>\langle\mu,f\rangle$. We describe an algorithm for finding such an f', in time polynomial in $|V|,|E|$.

Since all capacities are ones, one may assume that every integer multiflow is a set of pairwise edge-disjoint T-paths (which have unit weights). Let E^0 be the sets of residual (not used in the paths of f) edges of G. One may assume that each component of (V,E^0) contains at most one terminal; otherwise the problem is trivial. Then E^0 is representable as the union of pairwise edge-disjoint circuits. In what follows by a *decomposition* we mean a decomposition $\mathcal{D}'$ of G (or another inner Eulerian graph in question) into T-paths and circuits. The *μ-value* $\langle\mu,\mathcal{D}'\rangle$ of $\mathcal{D}'$ is the μ-value of the corresponding multiflow formed by the T-paths in $\mathcal{D}'$.

Let $\mathcal{D}$ be a decomposition including the initial f, and let $\overline{\nu}=\langle\mu,f\rangle=\langle\mu,\mathcal{D}\rangle$. The algorithm we develop handles $\mathcal{D}$ rather than f, attempting to transform $\mathcal{D}$ into a decomposition $\mathcal{D}'$ with a greater μ-value.

To simplify the algorithm description, we assume that each inner node of G has degree at most four. This leads to no loss of generality. Indeed, denote by $E(x)$ the set of edges incident to a node x, and suppose that $|E(x)|>4$ for some $x\in V-T$. Then we can transform the graph at x as follows. Partition each edge $e=xy$ into two edges xz_e and z_ey in series, and connect the nodes z_e, $e\in E(x)$, by a simple circuit C of new (residual) edges. For each pair $e,e'\in E(x)$ such that e,x,e' are consecutive elements of some member of $\mathcal{D}$, replace $xz_e,xz_{e'}$ by a single edge $z_ez_{e'}$, and then remove x. See Fig. 2 for an illustration. This transformation creates $|E(x)|$ nodes with degree four in place of x, and $\mathcal{D}$ is transformed, in a natural way, into a decomposition $\mathcal{D}'$ with the same μ-value in the resulting graph G'. Note that $\mathcal{D}'$ is not maximum in G'. (For otherwise take an optimal T-partition π' for G'. Then the nodes of C are entirely contained in one member of π'. Hence, π' induces an optimal T-partition π for G, and $\mathcal{D}$ is maximum.) Note also that any integer multiflow g' in G' can be easily transformed into an integer multiflow f' with the same μ-value in G. So if we succeed to find g' in G' with $\langle\mu,g'\rangle>\overline{\nu}$, it gives a multiflow f' in G with $\langle\mu,f'\rangle>\overline{\nu}$.

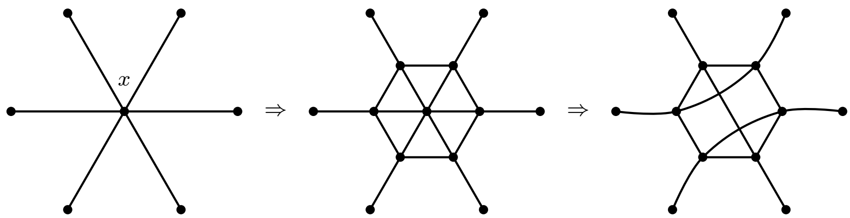

Fig. 2

Let W be the set of inner nodes of degree four in G. For each $x\in W$, $\mathcal{D}$ induces in a natural way a *bi-partition* $\rho(x)=\rho^{\mathcal{D}}(x)$ of the set $E(x)=\{e_1,e_2,e_3,e_4\}$ into two pairs, $\{e_1,e_2\}$ and $\{e_3,e_4\}$ say; we write $\rho(x)=\{e_1,e_2|e_3,e_4\}$. W.l.o.g., one may assume that each path or circuit in $\mathcal{D}$ is simple and that no path P in $\mathcal{D}$ contains a terminal as an intermediate node (otherwise split P at such a node into two T-paths; this does not decrease the μ-value). We say that such a $\mathcal{D}$ is *simple*. The non-maximality of f and the fact that no residual path connects different terminals imply that W is nonempty. Note that $\mathcal{D}$ is determined uniquely by the bi-partitions $\rho(x)$ for $x\in W$. Moreover, every set of bi-partitions $\rho'(x)$ at the elements $x\in W$ determines uniquely a (possibly non-simple) decomposition $\mathcal{D}_{\rho'}$ in which each circuit contains at most one terminal and each T-path meets T only at its ends.

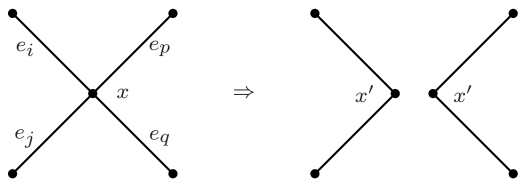

Fig. 3

Consider $x\in W$, and let $\rho(x)=\{e_1,e_2|e_3,e_4\}$. The *splitting operation* at x with respect to a bi-partition $\overline{\rho}=\{e_i,e_j|e_p,e_q\}$ of $E(x)$ (possibly $\overline{\rho}\neq\rho(x)$) transforms

G into a graph G' by replacing the end x of e_i, e_j by a new node x' and replacing the end x of e_p, e_q by another new node x''; see Fig. 3. The decomposition in G' corresponds to the (possibly non-simple) decomposition $\mathcal{D}' = \mathcal{D}_{\rho'}$, where $\rho'(x) = \overline{\rho}$ and $\rho'(y) = \rho(y)$ for all $y \in W - \{x\}$. We call such an operation

(i) an *augmenting* splitting if $\langle \mu, \mathcal{D}' \rangle > \overline{\nu}$;

(ii) a *feasible* splitting if $\langle \mu, \mathcal{D}' \rangle = \overline{\nu}$;

(iii) a *good* splitting if it is feasible and $\mathcal{D}'$ is still not maximum in the new graph G', i.e., $\nu(G', \mu) > \overline{\nu}$;

(iv) a *laminar* splitting if $\overline{\rho} = \rho(x)$, and *cross-splitting* otherwise.

In particular, the laminar splitting is, obviously, feasible (but needs not be good). Clearly to compute the value $\langle \mu, \mathcal{D}' \rangle$ and compare it with $\overline{\nu}$ is easy. If there is an augmenting splitting at some node $x \in W$, we immediately obtain an integer multiflow with a greater μ-value; so assume this is not the case. Also using the maximality check-up algorithm from the previous section, we can examine all feasible splittings at the nodes in W to seek a good one among them, and once a good splitting is found, we transform G into the graph G' whose set W' of inner nodes of degree four is smaller than W.

We will rely on the following fact.

Statement 3.1. *For $x \in W$, if all three splittings at x are feasible, then at least one of them is good.*

Proof. Take a decomposition $\mathcal{D}'$ of G with $\langle \mu, \mathcal{D}' \rangle > \overline{\nu}$ (it exists by Theorem 1.1 since f is not maximum in G), and let G' be obtained by the splitting at x w.r.t. the bi-partition $\rho' = \rho^{\mathcal{D}'}(x)$. Then $\nu(G', \mu) \geq \langle \mu, \mathcal{D}' \rangle > \overline{\nu}$ and, therefore, the given splitting is good (as it is feasible). ∎

In what follows we assume that no feasible splitting at any node is good. Let P and P' be the members of $\mathcal{D}$ passing $x \in W$. The above statement enables us to eliminate the following situations in which each of the two cross-splittings at x is feasible or even augmenting:

(3.1) (i) some of P, P' is a circuit;

(ii) P is a u–v path and P' is a z–w path and each of $\mu(uz) + \mu(vw)$ and $\mu(uw) + \mu(vz)$ is greater than or equal to $\mu(uv) + \mu(zw)$.

As before, we consider each path in f up to reversing, and for $u, v \in T$, denote by V_{uv} the domain of the flow f_{uv}. The impossibility of (3.1) implies:

(3.2) all members of $\mathcal{D}$ are (simple) T-paths;

(3.3) for $i = 1, 2$ and $j = 1, \ldots, r$, if $V_{s_i t_j}$ and V_{uv} with $s_i t_j \neq uv$ share an inner node, then uv is either $s_i s_{i'}$ or $t_j t_{j'}$; in particular, the flows $f_{s_i t_j}$ are pairwise openly disjoint, i.e., $V_{s_i t_j} \cap V_{s_{i'} t_{j'}} = \{s_i, t_j\} \cap \{s_{i'}, t_{j'}\}$ for $(i,j) \neq (i',j')$.

Note that at least one flow $f_{s_it_j}$ is nonzero (otherwise f is, obviously, maximum). Also f contains at least one s_1–s_2 or t_i–t_j path (otherwise (3.3) implies that $W=\emptyset$). Until now we have tried to get the desired augmentation or reduce the set W by applying a single splitting operation, and (3.2)–(3.3) expose all we are able to obtain on this way. However, we can rearrange the decomposition $\mathcal{D}$ more globally by combining several splittings at once in order to get a situation as in (3.1) and then make a crucial splitting.

For example, suppose that f contains an s_1–t_1 path $P=x_0\dots x_k$, an s_1–t_2 path $P'=y_0\dots y_p$ and an s_1–s_2 path $P''=z_0\dots z_q$ such that P'' meets both P and P' at intermediate nodes, $x_i=z_j=:x$ and $y_{i'}=z_{j'}=:y$ for $j<j'$ say. Then no single splitting at x or at y is good. Nevertheless, $P\cup P'\cup P''$ is decomposed into the s_1–s_2 path $Q=y_0\dots y_{i'}z_{j'+1}\dots z_q$, the t_1–t_2 path $Q'=x_k\dots x_iz_{j+1}\dots z_{j'}y_{i'+1}\dots y_p$ and the circuit $C=x_0\dots x_iz_{j-1}\dots z_0$. Hence, replacing P,P',P'' in $\mathcal{D}$ by Q,Q',C makes $\mathcal{D}'$ with $\langle\mu,\mathcal{D}'\rangle=\langle\mu,\mathcal{D}\rangle$, and now a good splitting at x becomes possible since x belongs to a T-path and a circuit.

In general, the method of rearranging f is as follows. Assume that $f_{s_1t_1}$ is nonzero. We construct a minimal set $L=L_1=L(s_1,t_1)$ such that $V_{s_1t_1}\subseteq L$ and each s_1–s_2 or t_1–t_p path of f meets the cut $\delta(L)$ at most once. Such an L is unique and is constructed by use of a labelling method. Initially, set $L:=V_{s_1t_1}$. Then increase L step by step by the following rule:

(3.4) choose an s_1–s_2 or t_1–t_p path $P=x_0\dots x_k$ in f such that for some $0<i<j<k$, $x_i\notin L$ and $x_j\in L$, and update $L:=L\cup\{x_0,\dots,x_j\}$.

The process terminates when L cannot be increased by (3.4). Clearly $L\cap T=\{s_1,t_1\}$. Let $U_1=\{s_1s_2,t_1t_2,\dots,t_1t_r\}$ and $\overline{U}_1=U-(U_1\cup\{s_1t_1\})$. Suppose that there is a node $x\in L-\{s_1,t_1\}$ such that

(3.5) x belongs to V_{zw} for some $zw\in\overline{U}_1$.

We show that the submultiflow g in f that consists of the flows $f_{s_1t_1}$ and f_{uv}, $uv\in U_1$, can be rearranged within L so as to create an s_1–t_1 path P containing x. Then P and the path in f_{zw} containing x, where zw is as in (3.5), are in situation (3.1)(ii) and, therefore, one can apply an augmenting or good splitting at x. The task of finding such a rearrangement is reduced to the multiflow demand problem (defined in the Introduction) for the graph $\Gamma=(F,Z)$, demand pairs D and demands d on the values of flows connecting pairs in D, where:

(i) Γ is formed by the nodes and edges occurring in g or in T;

(ii) D consists of the pairs s_1x, xt_1, s_1t_1 and the pairs in U_1;

(iii) $d(s_1x)=d(xt_1)=1$, $d(s_1s_2)=|f_{s_1s_2}|-1$, and $d(uv)=|f_{uv}|$ for $uv\in U_1$ ($|f_{uv}|$ denotes the number of paths in f_{uv}).

Statement 3.2. *The above demand problem has an integer solution, i.e., there exist pairwise edge-disjoint T'-paths $Q_1,\dots,Q_q$ in Γ such that each pair uv in D is connected by $d(uv)$ of these paths, where $T'=T\cup\{x\}$.*

Proof. For $X \subseteq F$, define d_X as in (1.7), and let $\Delta(X) = |\delta^\Gamma(X)| - d_X$. It is easy to see that $\Delta(X)$ is an even integer for any X. Also each pair in D meets at least one of s_1, t_1, i.e., the graph induced by D is the union of two stars. Hence, the problem has an integer solution provided that the cut condition holds, i.e., if $\Delta(X) \geq 0$ for all $X \subseteq F$ (by a simple reduction (see [1]) from the two-star commodity demand problem to the two commodity problem and by a theorem of Rothschield and Whinston concerning the latter [16]). Suppose that $\Delta(X) < 0$ for some X; let for definiteness $s_1 \in X$. Considering the feasible multiflow g in Γ, we observe that $\Delta(X) < 0$ is possible only if $s_1, t_1 \in X$ and $x \notin X$. One may assume that $X \cap T' = \{s_1, t_1\}$ (for if $v \in X$ for some $v \in T - \{s_1, t_1\}$, then, obviously, $\Delta(X - \{v\}) \leq \Delta(X)$).

Let $\overline{d} = \sum(|f_{uv}| : uv \in U_1)$. Then $|\delta^\Gamma(X)| \geq \overline{d}$ and $\overline{d} = d_X - 2$. Therefore, the evenness of $\Delta(X)$ implies $\Delta(X) = -2$, whence $|\delta^\Gamma(X)| = \overline{d}$. The latter equality means that $V_{s_1t_1} \subseteq X$ and each u–v path in f with $uv \in U_1$ meets $\delta^\Gamma(X)$ precisely once. Then no node outside X can be labelled by rule (3.5). So $x \notin L$; a contradiction. ∎

Let Q_i's be as in Statement 3.2, and let for definiteness Q_1 and Q_2 be the paths from s_1 to x and from x to t_1, respectively. Then the multiflow g' formed by the paths $Q_3, \ldots, Q_q$ and the s_1–t_1 path being the concatenation of Q_1 and Q_2 satisfies $\langle \mu, g' \rangle = \langle \mu, g \rangle$. We replace in f the part g by g', obtaining situation (3.1)(ii), and proceed with an augmenting or good splitting at x.

If there is no x as in (3.5), we consecutively construct the labelled sets $L_2 = L(s_1, t_2), \ldots, L_r = L(s_1, t_r)$ in a similar way (letting $L_i = \{s_1, t_i\}$ if $f_{s_1t_i}$ is empty). At least one set $L_i - \{s_1, t_i\}$ must meet V_{uv} with uv different from s_1t_i, s_1s_2 and t_it_q, $q \in \{1, \ldots, r\} - \{i\}$. For otherwise, in view of (3.3), $L_i \cap L_j = \{s_1\}$ for all $i \neq j$; then the sets $\{s_1\}$, $L_i - \{s_1\}$ $(i = 1, \ldots, r)$ and $V - (L_1 \cup \ldots \cup L_r)$ are easily shown to form an optimal T-partition, implying that f is maximum.

The above demand problem can be solved in time polynomial in $|V|, |E|$ (e.g., by the method behind the proof in [16]). Since the number of splittings we apply does not exceed $|V|$, the whole time needed to find an integer augmentation of f is polynomial.

4. Scaling method

In this section we put together the above arguments to design a polynomial algorithm for solving (1.6) and the corresponding integer multiflow problem (1.3). As before, μ is the path metric of the graph $K_{2,r}$ and the capacity function c on the edges of $G = (V, E)$ is assumed to be inner Eulerian.

At the high level, the algorithm applies a capacity scaling approach and consists of *big* (or scaling) iterations. The number of these iterations is equal to the size $||c||$ of the largest capacity in binary notation.

More precisely, let I be $\lceil \log_2(||c||+1)\rceil$. For $i=0,\dots,I$ and $e\in E$, define the truncated capacity $c_i(e)=\lfloor c(e)/2^{I-i}\rfloor$. Then $c_0=0$ and $c_I=c$. In the input of ith big iteration, there is a maximum *half-integer* multiflow g_{i-1} for G,c_{i-1},μ (letting g_0 be zero multiflow), and the goal is to find a maximum half-integer multiflow g_i for c_i. (The reason why we are forced to deal with half-integer multiflows is that c_i needs not be inner Eulerian.) The final, Ith, big iteration will find a maximum *integer* multiflow for G,c,μ along with a minimum $(2,r)$-metric.

We describe ith iteration, $i<I$. It considers $c'=2c_i$ as the capacity function and $g'=4g_{i-1}$ as the initial multiflow (i.e., the weights of all paths in g_{i-1} are increased by a factor of four). Then c' is inner Eulerian and g' is c'-admissible and integral. Moreover, we observe that

$$\tau(c')-\langle\mu,g'\rangle\le 4|E|, \tag{4.1}$$

where $\tau(c')$ stands for $\tau(G,c',\mu)$. Indeed, let m be a minimum $(2,r)$-metric for c_{i-1}. Then g' and m are optimal for $4c_{i-1}$, i.e., $\langle\mu,g'\rangle=(4c_{i-1})\cdot m$. For each $e\in E$, we have $m(e)\le 2$ and $c'(e)=2c_i(e)\le 4c_{i-1}(e)+2$. Therefore, $\tau(c')\le c'\cdot m\le(4c_{i-1})\cdot m+4|E|$, yielding (4.1).

Thus, g' is nearly optimal for c', and it takes at most $4|E|$ integer augmentations to transform g' into a maximum integer multiflow for c'. However, the integer augmentation algorithm from Section 3 is sensitive to the capacity values (it is polynomial only for small capacities and pseudo-polynomial in general). To this reason, every time we turn to this algorithm, we first transform c' and the current integer multiflow g as follows.

Let $g=(P_1,\dots,P_k;\lambda_1,\dots,\lambda_k)$, where all λ_i's are positive integers. One may assume that each flow g_{uv} in g consists of at most $|E|$ paths (for we can rearrange g_{uv}, if needed, using standard flow decomposition techniques [7]). Then k is $O(|T|^2|E|)$. We replace g by the multiflow h consisting of the same paths $P_1,\dots,P_k$ but taken with weight one each. Also we represent the residual capacities $\Delta(e)=c'(e)-g^e$ as $\Delta=\alpha_1\chi^{C_1}+\ldots+\alpha_q\chi^{C_q}$, where $q\le|E|$, each α_i is a positive integer and each C_i is a circuit in G. We replace c' by $c''=\chi^{P_1}+\ldots+\chi^{P_k}+\chi^{C_1}+\ldots+\chi^{C_q}$. Then $||c''||$ is only $O(|T|^2|E|)$ and, therefore, replacing each edge e by $c''(e)$ parallel edged makes an inner Eulerian multigraph whose size is polynomial in $|V|,|E|$. By Corollary 2.2, h is not maximum for c'' if and only if g is not maximum for c'. Moreover, an integer augmentation of h by use of the algorithm in Section 3 determines an integer augmentation of g in a natural way.

Summing up the above arguments, we conclude that each big iteration is performed in strongly polynomial time and finds a maximum integer multiflow g for $2c_i$, yielding the maximum half-integer multiflow $g_i=\frac{1}{2}g$ for c_i. The final, Ith, iteration is applied to the inner Eulerian $c_I=c$ and integer multiflow $g'=2g_{I-1}$ (instead of $2c_I$ and $4g_{I-1}$) and finds a maximum integer multiflow f for the initial G,c,μ. Also the last application of the algorithm from Section 2 constructs a minimum $(2,r)$-metric m. Thus, the total time of our algorithm is polynomial in $|V|, |E|$ and linear in $\log ||c||$, as required.

5. Algorithm for sparse frames

We will use some properties of modular graphs. A graph $H = (T,U)$ is called *modular* if each three nodes $v_0, v_1, v_2 \in T$ have a *median*, i.e., a node $z \in T$ such that $\mathrm{d}^H(v_i z) + \mathrm{d}^H(z v_j) = \mathrm{d}^H(v_i v_j)$ for all $0 \le i < j \le 2$. If every isometric subgraph of H is modular, then H is said to be *hereditary modular*. Any modular graph is bipartite.

Bandelt [3] proved the following theorem: a bipartite graph H has no isometric k-circuit with $k \ge 6$ if and only if H is hereditary modular. Thus, the frames (figured in Theorem 1.2) are precisely the orientable hereditary modular graphs. Modular graphs have the following property, which is a consequence of a result in [2]:

(5.1) for any orbit Q of a modular graph H and any $u, v \in T$, if P is a shortest u–v path and P' is a u–v path, then $|P \cap Q| \le |P' \cap Q|$.

For a nonnegative function ℓ on the edges of a graph H, we denote by $\mathrm{d}^{H,\ell}$ the path metric for (H, ℓ), i.e., $\mathrm{d}^{H,\ell}(xy)$ is the minimum ℓ-length $\ell(P) = \sum(\ell(e) : e \in P)$ of a path P connecting nodes x and y in H. An *ℓ-shortest* path is a path shortest for the metric $\mathrm{d}^{H,\ell}$. We say that ℓ is *orbit-invariant* if it is constant within each orbit of H. Consider a modular graph $H = (T,U)$, and let $Q_1, \dots, Q_k$ be the orbits of H. From (5.1) it follows that

(5.2) for any nonnegative orbit-invariant function ℓ on U, each shortest path in H is ℓ-shortest.

Indeed, let $h_i = \ell(e)$ for $e \in Q_i$. For two u–v paths P and P' in H, we have $\ell(P) = h_1 n_1 + \ldots + h_k n_k$ and $\ell(P') = h_1 n'_1 + \ldots + h_k n'_k$, where $n_i = |P \cap Q_i|$ and $n'_i = |P' \cap Q_i|$. If P is shortest, then $n_i \le n'_i$ for each i (by (5.1)) implies $\ell(P) \le \ell(P')$ since ℓ is nonnegative.

For $i = 1, \dots, k$, define $\ell_i = \chi^{Q_i}$ and let $\mu_i = \mathrm{d}^{H,\ell_i}$.

Statement 5.1. $\mathrm{d}^H = \mu_1 + \ldots + \mu_k$.

Proof. Consider a shortest u–v path P in H, and let $|P|$ denote the number of its edges. By (5.2), P is ℓ_i-shortest for $i = 1, \dots, k$. Therefore, $|P| = \ell_1(P) + \ldots + \ell_k(P)$ implies $\mathrm{d}^H(uv) = \mu_1(uv) + \ldots + \mu_k(uv)$. ∎

For $i = 1, \dots, k$, let $H_i = (T_i, U_i)$ be the graph in Definition 1.3. Define π_i to be the partition of T formed by the node sets of components of the graph $(T, U - Q_i)$. We formally identify each node of H_i with some node in the corresponding member of π_i. Then $T_i \subseteq T$ and μ_i is a 0-extension of $\widehat{\mu}_i = \mathrm{d}^{H_i}$ to T.

Consider a graph $G = (V,E)$ with $V \supseteq T$ and a capacity function c on E. Let $G_i = (V_i, E)$ be the graph (with possible parallel edges and loops) obtained from G

by shrinking each set in π_i into the corresponding node of H_i. Every 0-extension m' of μ_i to V one-to-one corresponds, in a natural way, to a 0-extension $\widehat{m}'$ of $\widehat{\mu}_i$ to V_i, and we have $c\cdot m'=c\cdot\widehat{m}'$. Let $\tau=\tau(G,c,\mathrm{d}^H)$ and $\tau_i=\tau(G,c,\mu_i)$.

Let m be a minimum 0-extension for G,c,d^H. From Statement 5.1 it follows that m is represented as $m=m_1+\ldots+m_k$, where each m_i is a 0-extension of μ_i to V. This implies

$$\tau \ge \tau_1 + \ldots + \tau_k. \tag{5.3}$$

Moreover, (5.3) turns into equality for any frame H. Indeed, for $i=1,\ldots,k$, take a 0-extension m_i of μ_i to V with $c\cdot m_i=\tau_i$. By Statement 5.1, $m=m_1+\ldots+m_k$ is an extension of d^H to V, whence $c\cdot m\ge\tau^*$ $(=\tau^*(G,c,\mathrm{d}^H))$. This implies $\tau^*\le\tau_1+\ldots+\tau_k$, yielding $\tau\le\tau_1+\ldots+\tau_k$, in view of $\tau=\tau^*$.

The above arguments prompt an approach to solve (1.1) for $\mu=\mathrm{d}^H$, where H is a sparse frame. First we find the numbers τ_i by solving (1.1) for each $G_i,c,\widehat{\mu}_i$; the sum of τ_i's amounts to τ. (Note that if H_i is $K_{1,r}$, the problem for $G_i,c,\widehat{\mu}_i$ is reduced to r minimum cut computations. In fact, one can show that $r=1$ is only possible but this is not important for us.) In order to find the desired minimum 0-extension of d^H to V, we can apply a method suitable for arbitrary minimizable metrics (see [13]). More precisely, choose a node $x\in V-T$ and a terminal $t\in T$ and compute $\tau'=\tau(G',c,\mathrm{d}^H)$, where G' is obtained from G by identifying x and t. If $\tau'=\tau$, we replace G by G'. And if $\tau'>\tau$, choose another terminal t' to be identified with x and do similarly, and so on (since d^H is minimizable, at least one choice of a terminal t for the x gives $\tau'=\tau$). Then repeat the procedure with a next node $x'\in V-T$. Eventually, after at most $|V-T||T|$ iterations, we obtain a graph $\widetilde{G}=(\widetilde{V},E)$ with $\widetilde{V}=T$ and $\tau(\widetilde{G},c,\mathrm{d}^H)=\tau$, which determines the minimum 0-extension for G,c,d^H in an obvious way.

However, we can suggest a more efficient method in which the problem with the same μ_i is solved only once. We say that a metric m on V is *cyclically even* if $m(xy)+m(yz)+m(zx)$ is an even integer for any $x,y,z\in V$ (in particular, m is integral). An extension m of d^H to V is called *tight* if each two $x,y\in V$ belong to an m-shortest T-path, i.e., $m(sx)+m(xy)+m(yt)=\mathrm{d}^H(st)$ for some $s,t\in T$.

As before, we first find an optimal 0-extension m_i of μ_i to V for each i and put $m=m_1+\ldots+m_k$. Each m_i is cyclically even (as H_i is bipartite); therefore, m is cyclically even as well. Next we find a cyclically even tight extension $m'\le m$ of d^H (note that m needs not be tight). Initially, set $m'=m$. Choose $x,y\in V$ such that $m'(xy)\ge 2$ and

$$\Delta(xy) := \min\{m'(sx)+m'(xy)+m'(yt)-\mathrm{d}^H(st)\ :\ s,t\in T\} > 0.$$

Since m' is cyclically even, $\Delta(xy)$ is even. Decrease the length of xy by the maximum even integer exceeding neither $\Delta(xy)$ nor $m'(xy)$. For the resulting function ℓ (not necessarily a metric), update $m' := \mathrm{d}^{K,\ell}$, where $K=(V,E_V)$ is the complete graph on V. Then any T-path P in K satisfies $\ell(P) \geq \mathrm{d}^H(st)$, where s,t are the ends of P, whence the new m' is again an extension of d^H. Also one can see that m' is cyclically even. Repeat the procedure for a next pair $x',y' \in V$, and so on. After $O(|V|^2)$ iterations we obtain a cyclically even extension m' of d^H such that for any $x,y \in V$,

(5.4) if $m'(xy) \geq 2$, then $\Delta(xy)=0$.

Statement 5.2. $\Delta(xy)=0$ *for all* $x,y \in V$ *(assuming* $|T| \geq 2$*).*

Proof. By (5.4), it suffices to examine x,y with $m'(xy) \leq 1$. Let $m'(xy)=1$. Choose $z \in V$ such that both $m'(zx), m'(zy)$ are nonzero (if no such z exists, then, in view of $|T| \geq 2$, $m'(sx)=m'(ty)=0$ for some $s,t \in T$, whence $\Delta(xy)=0$). Since m' is cyclically even and $m'(xy)=1$, either $m'(zx)+m'(xy)=m'(zy)$ or $m'(zy)+m'(yx)=m'(zx)$. For definiteness, assume the former. Then $m'(zy) \geq 2$. Therefore, there exist $s,t \in T$ such that $m'(sz)+m'(zy)+m'(yt)=\mathrm{d}^H(st)$. This implies $m'(sx)+m'(xy)+m'(yt)=\mathrm{d}^H(st)$, yielding $\Delta(xy)=0$. In case $m'(xy)=0$, the result follows from $\Delta(xz)=0$ for $z \in V$ with $m'(xz)>0$. ∎

Thus, m' is tight. Also $m' \leq m$ implies $c \cdot m' \leq c \cdot m$, whence $c \cdot m' = \tau$. We assert that m' is a 0-extension of d^H and, therefore, m' is an optimal solution to (1.1); moreover, a similar property is true for an arbitrary frame H. The proof relies on a result concerning the tight spans for frames. More precisely, as shown in [8] (see also [6]), every (reasonable) metric space (X,d) possesses a unique positive tight extension $(\mathcal{X},\delta)$ such that any tight extension (X',d') of (X,d) is isometrically embeddable in $(\mathcal{X},\delta)$, in the sense that there exists a mapping $\gamma: X' \to \mathcal{X}$ with the identity on X satisfying $d'(xy)=\delta(\gamma(x)\gamma(y))$ for all $x,y \in X'$. Such an $(\mathcal{X},\delta)$ is called the *tight span* (or injective envelope, or T_X-space) of (X,d). [11] gives an explicit combinatorial construction for the tight span of the path metric of any frame. The proof of validity of that construction involves the following key result.

Statement 5.3 [11]**.** *Let* $H=(T,U)$ *be a frame and let* m' *be a tight extension of* d^H *to* $V \supseteq T$*. For each* $x \in V$*, at least one of the following is true:*

(i) $m'(vx)=0$ *for some* $v \in T$*;*

(ii) $m'(ux)+m'(xv)=1$ *for some edge* $uv \in U$*;*

(iii) $m'(v_0x)+m'(xv_2)=m'(v_1x)+m'(xv_3)=2$ *for some 4-circuit* $C=v_0v_1v_2v_3v_0$ *of* H*.*

Using this fact, we observe that for each $x \in V$, there exists $t \in T$ with $m'(tx)=0$. This is immediate in cases (i) and (ii) of Statement 5.3. So assume that we are in case (iii) and that $m'(v_ix)>0$ for $i=0,1,2,3$. Then $m'(v_ix)=1$ for each i. But $m'(v_0x)+m'(xv_1)+m'(v_1v_0)=1+1+1=3$, contrary to the fact that m' is cyclically even. Thus, m' is indeed a 0-extension.

The above arguments give a "purely combinatorial" algorithm for solving (1.1) with the path metric $\mu = \mathrm{d}^H$ of a sparse frame H. This algorithm involves $O(|T|)$ minimum $(2,r)$-metric computations and runs in time polynomial in $|V|,|E|$ and linear in $\log ||c||$.

In fact, this algorithm solves (1.1) for any metric μ of the form $\mathrm{d}^{H,\ell}$, where $H=(T,U)$ is a sparse frame and ℓ is a nonnegative orbit-invariant function on U. This relies on the following fact (which can be shown by use of (5.2); cf. also [10]).

Statement 5.4. *Let m be a minimum 0-extension for G,c,d^H, where H is a frame. Let ℓ be a nonnegative orbit-invariant function on the edges of H, and let $\mu=\mathrm{d}^{H,\ell}$. Then m is a minimum 0-extension for G,c,μ.*

6. Concluding remarks and open questions

The method for solving the minimum $(2,r)$-metric problem developed in Sections 2–4 has a weakly polynomial time bound, and we do not see how to get rid of scaling operations in order to obtain a "combinatorial" strongly polynomial algorithm for this problem. Another open question: can one construct a "combinatorial" polynomial algorithm for the minimum 0-extension problem in a general case of minimizable metrics μ?

Next, extending a result in [5] on the intractability of the minimum 3-terminal cut problem (i.e., (1.1) with $\mu=\mathrm{d}^{K_3}$), [11] proves that for a fixed $\mu=\mathrm{d}^H$, problem (1.1) is strongly NP-hard if H is non-modular or non-orientable (e.g., if $H=C_5$ or $K_{3,3}$). On the other hand, (1.1) is solvable in strongly polynomial time for each minimizable metric μ. There is one more polynomial case, namely, when μ is representable as $\mu=\lambda_1\mu_1+\ldots+\lambda_k\mu_k$, where $\lambda_1,\ldots,\lambda_k \geq 0$, each μ_i is the cut metric on T corresponding to a bi-partition $\{A_i,B_i\}$ of T, and the family $\mathcal{F}=\{A_1,\ldots,A_k,B_1,\ldots,B_k\}$ satisfies the Helly property, i.e., any subfamily of $\mathcal{F}$ has a nonempty intersection provided that each two members of this subfamily meet. (When μ is d^H, this is equivalent to the property that H is a median graph, i.e., any three nodes of H have precisely one median [15].) A strongly polynomial algorithm for this case is given in [4] (see also [11] for a simple strongly polynomial algorithm based on a reduction to k minimum cut computations and uncrossing techniques).

Yet, there remain metrics $\mu=\mathrm{d}^H$ for which the complexity status of (1.1) is unknown. Let $H=(T,U)$ be modular and orientable but not necessarily a frame, and let $Q_1,\ldots,Q_k$ be the orbits of H. Then d^H is $\mu_1+\ldots+\mu_k$, where μ_i is the corresponding 0-extension of $\mathrm{d}^{H/(U-Q_i)}$ to T, as explained in Section 5. We conjecture that if each $H/(U-Q_i)$ is a frame, then $\tau=\tau_1+\ldots+\tau_k$ (cf. (5.3)) holds for any G,c, and (1.1) is in P (a stronger conjecture is that similar properties take place for any orientable modular graph H).

In this case, each subproblem with G,c,μ_i is solvable in strongly polynomial time, and one can construct a cyclically even tight extension m' of d^H to V satisfying $c\cdot m' \le \tau_1+\ldots+\tau_k$ by use of the method in Section 5. If m' is a 0-extension, then m' is an optimal solution to (1.1), and (5.3) turns into equality. However, m' needs not be a 0-extension. For example, suppose that H includes a cube $H'=(T',U')$ as an isometric subgraph, and let $A\subset T'$ be a stable set of four nodes. Then d^H has a cyclically even tight extension m on $V=T\cup\{x\}$ such that $m(xv)=1$ for each $v\in A$ and $m(xv)=2$ for each $v\in T'-A$; so m is not a 0-extension in general.

There is a special case when the sum m of minimum 0-extensions m_i for the partial problems with G,c,μ_i is automatically a minimum 0-extension for the whole problem. More precisely, let $H=(T,U)$ be the Cartesian product of modular graphs $H^i=(T^i,U^i)$, $i=1,\ldots,n$, i.e., $T=T^1\times\ldots\times T^n$ and nodes $(x_1,\ldots,x_n)$ and $(y_1,\ldots,y_n)$ are connected by an edge in H if and only if they differ at exactly one component, $x_i\ne y_i$ say, and $x_iy_i\in U^i$. For $i=1,\ldots,n$, let μ_i be the 0-extension of d^{H^i} to T, defined by $\mu_i(xy)=\mathrm{d}^{H^i}(x_iy_i)$ for $x,y\in T$. Then $\mathrm{d}^H=\mu_1+\ldots+\mu_n$ and H is modular.

Statement 6.1. *(i) H is orientable if each H^i is orientable.*

(ii) For any 0-extensions m_i of μ_i to V $(i=1,\ldots,n)$, $m=m_1+\ldots+m_n$ is a 0-extension of d^H to V.

Proof. Feasible orientations of the graphs H^i induce an orientation of H in a natural way. To see that this orientation is feasible, consider a 4-circuit $C=xyzvx$ in H. One can see that either for some i, $x_iy_iz_iv_ix_i$ is a 4-circuit in H^i, or for some i,j, $x_iy_i=v_iz_i$ is an edge of H^i and $x_jv_j=y_jz_j$ is an edge of H^j. In both cases, the orientation of C is feasible. Finally, for $x\in V$ and $i=1,\ldots,n$, take $t_i\in T^i$ such that $m_i(xt_i)=0$. Then $t=(t_1,\ldots,t_n)$ satisfies $m(xt)=m_1(xt_1)+\ldots+m(xt_n)=0$, yielding (ii). ∎

By (ii) in this statement and (5.3), $\tau=\tau_1+\ldots+\tau_n$ holds for any G,c. Moreover, when H is the Cartesian product of frames, we obtain a strongly polynomial algorithm for (1.1) with $\mu=\mathrm{d}^H$.

References

[1] G. M. Adelson-Velsky, E. A. Dinitz, and A.V. Karzanov: *Flow Algorithms,* Nauka, Moscow, 1975, in Russian.

[2] H.-J. Bandelt: Networks with Condorcet solutions, *European J. Oper. Res.,* **20** (1985), 314–326.

[3] H.-J. Bandelt: Hereditary modular graphs, *Combinatorica,* **8** (2) (1988), 149–157.

[4] V. Chepoi: A multifacility location problem on median spaces, *Discrete Applied Math.,* **64** (1996), 1–29.

[5] E. Dalhaus, D. S. Johnson, C. Papadimitriou, P. Seymour, M. Yannakakis: The complexity of the multiterminal cuts, *SIAM J. Comput.,* **23** (4) (1994), 864–894.

[6] A. W. M. Dress: Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups, *Advances in Mathematics,* **53** (1984), 321–402.

[7] L. R. Ford and D. R. Fulkerson: *Flows in networks,* Princeton Univ. Press, Princeton, NJ, 1962.

[8] J. Isbell: Six theorems about metric spaces, *Comment. Math. Helv.,* **39** (1964), 65–74.

[9] A. V. Karzanov: Half-integral five-terminus flows, *Discrete Applied Math.,* **18** (3) (1987), 263–278.

[10] A. V. Karzanov: Metrics with finite sets of primitive extensions, to appear in *Annals of Combinatorics.*

[11] A. V. Karzanov: Minimum 0-extensions of graph metrics, *European J. Combinatorics,* **19** (1998), 71–101.

[12] A. V. Karzanov and M. V. Lomonosov: Systems of flows in undirected networks, in: *Mathematical Programming etc.* (Inst. for System Studies, Moscow, 1978, iss. 1) 59–66, in Russian.

[13] A. V. Karzanov and Y. Manoussakis: Minimum $(2,r)$-metrics and integer multiflows, *European J. Combinatorics,* **17** (1996), 223–232.

[14] M. V. Lomonosov: Combinatorial approaches to multiflow problems, *Discrete Applied Math.,* **11** (1) (1985), 1–94.

[15] H. M. Mulder and A. Schrijver: Median graphs and Helly hypergraphs, *Discrete Math.*, **25** (1979), 41–50.

[16] B. Rothschield and A. Whinston: On two-commodity network flows, *Operations Research*, **14** (1966), 377–387.

[17] B. C. Tansel, R. L. Fransis, T. J. Lowe: Location on networks: a survey I, II, *Management Sci.,* **29** (1983), 482–511.

Alexander V. Karzanov

Institute for System Analysis
9, Prospect 60 Let Oktyabrya,
117312 Moscow, Russia
sasha@cs.isa.ac.ru