Max-Weight Integral Multicommodity Flow in Spiders and High-Capacity Trees

Jochen Könemann[1], Ojas Parekh[2], and David Pritchard[1]

[1] University of Waterloo
jochen@math.uwaterloo.ca, dagpritchard@math.uwaterloo.ca
Dept. of Combinatorics & Optimization, 200 University Avenue West,
Waterloo, Ontario, Canada, N2L 3G1. Fax: 519-725-5441
[2] Emory University
ojas@mathcs.emory.edu
Math/CS Department, 400 Dowman Dr., Suite 401,
Atlanta, GA, USA, 30322. Fax: 404-727-5611

Abstract. We consider the max-weight integer multicommodity flow problem in trees. In this problem we are given an edge-capacitated tree and weighted pairs of terminals, and the objective is to find a max-weight integral flow between terminal pairs subject to the capacities. This problem was shown to be APX-hard by Garg, Vazirani and Yannakakis [Algorithmica, 1997], and a 4-approximation was given by Chekuri, Mydlarz and Shepherd [ACM Trans. Alg., 2007]. Some special cases are known to be solvable in polynomial time, including when the graph is a star (via b-matching) or a path.
First, when every edge has capacity at least $\mu \geq 2$, we use *iterated relaxation* to obtain an improved approximation ratio of $\min\{3, 1+4/\mu+6/(\mu^2-\mu)\}$. We show this ratio bounds the integrality gap of the natural LP relaxation. A complementary hardness result yields a $1+\Theta(1/\mu)$ threshold of approximability (if $\mathsf{P} \neq \mathsf{NP}$). Second, we extend the range of instances for which exact solutions can be found efficiently. When the tree is a *spider* (i.e. if only one vertex has degree greater than 2) we give a polynomial-time algorithm to find an optimal solution, as well as a polyhedral description of the integer hull of all feasible solutions.

1 Introduction

In the max-weight integral multicommodity flow problem (WMCF), we are given an undirected *supply* graph $G = (V, E)$, terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$ where $s_i, t_i \in V$, non-negative weights $w_1, \ldots, w_k$, and non-negative integral edge-capacities c_e for all $e \in E$. The goal is to simultaneously route integral s_i-t_i flows of value y_i subject to the capacities, so as to maximize the weight $\sum w_i y_i$.

The single-commodity version ($k = 1$) of WMCF is well-known to be solvable in polynomial time. If we drop the integrality restriction the problem can be solved in polynomial time via linear programming for any k. However, when integrality is required, even the 2-commodity unit-capacity, unit-weight version is NP-complete — see Even, Itai, and Shamir [12]. Moreover, recent results of Andrews et al. [1, 2] show that WMCF is polylogarithmically hard to approximate unless $\mathsf{NP} \subset \mathsf{ZPTIME}(n^{\text{polylog}(n)})$, even with unit capacities and weights.

One avenue investigated by several papers, in which polylog-inapproximability does not apply, is the special case where the supply graph G is a tree, which we denote by WMCFT. Garg, Vazirani and Yannakakis [15] considered the unit-weight case and showed APX-hardness even if G's height is at most 3 and all capacities are 1 or 2; but on the positive side, they gave a 2-approximate polynomial-time primal-dual algorithm. Techniques of Garg et al. show that WMCFT can be solved in polynomial time when G has unit capacity (using dynamic programming and matching) or is a star (this problem is essentially equivalent to b-matching). The case where G is a path (so-called *interval packing*) is also polynomial-time solvable [5, 7, 17], e.g. by linear programming since the natural LP has a totally unimodular constraint matrix. For general WMCFT, without restrictions on capacities or weights, Chekuri, Mydlarz and Shepherd [7] gave a 4-approximation algorithm, and this remains the best ratio known to date.

Results. Throughout the paper, we use μ to denote the minimum capacity of any edge in the WMCFT instance. In the first part of the paper we use *iterated rounding/relaxation* to develop improved approximation ratios for WMCFT when μ is suitably large. Iterated relaxation yields an integral solution with optimal value or better but exceeding edge capacities by up to 2 (additively). This resolves a problem stated in Chekuri et al. [7]; in their words we prove "the c-relaxed integrality gap is 1" for $c = 2$ whereas they could not prove it for any constant c.[3]

The best prior approach for WMCFT with high capacities was *randomized LP rounding* [22], which gives a $1 + O(\sqrt{\log |V|/\mu})$-approximation algorithm. An improvement can be obtained by plugging the iterated rounding result into Chekuri et al. [7, Cor. 3.5], giving a $1 + O(1/\sqrt{\mu})$ approximation. Our first main result gives an even better result for large capacities and also improves on the previous best ratio of 4 when $\mu \geq 2$.

Theorem 1 *For WMCFT, there are polynomial-time algorithms achieving (a) approximation ratio 3 for $\mu \geq 2$, and (b) approximation ratio $(1 + 4/\mu + 6/(\mu^2 - \mu))$ for general μ.*

A slight modification of Garg et al.'s hardness proof shows that for some $\epsilon > 0$, for all $\mu \geq 2$, it is NP-hard to approximate WMCFT within a ratio of $1 + \epsilon/\mu$; we detail this modification in Appendix A. Thus (if $\mathsf{P} \neq \mathsf{NP}$) the ratio in Theorem 1(b) is tight up to the constant in the $\frac{1}{\mu}$ term.

Our methodology for Theorem 1 is to decrease the additively-violating solution towards feasibility, without losing too much weight. Part (a) uses an argument of Cheriyan, Jordán and Ravi [8]. Part (b) relies on an auxiliary *covering* problem; every feasible cover, when subtracted from

[3] To say that the c-relaxed integrality gap is 1 for an LP means that, where OPT is the LP's optimum, there is an integral solution of value at least OPT but violating each constraint by up to $+c$. E.g. when $c = 0$ this means the LP has an integral optimum.

the +2-violating WMCFT solution, results in a feasible WMCFT solution. An approach due to Jain [18] shows that iterated LP rounding, applied to the auxiliary problem, leads to a provably low-weight integral solution for the covering problem. A crucial fact in obtaining Theorem 1(b) is that the approximation guarantee of Jain's approach is relative to the optimal *fractional* solution of the natural LP.

Our second result maps out more of the landscape of "easy" and "hard" WMCFT instances. An *all-ror* instance is one in which, for some choice of root vertex, each commodity path either goes through the root or is *radial*. (A path is radial if one of its endpoints is an ancestor of the other with respect to the root.) For example, every instance of WMCFT in which G is a spider is an all-ror instance.

Theorem 2 *All-ror WMCFT instances can be solved in strongly polynomial time.*

One way to view this result is as an efficient solution for a common generalization of b-matching and interval packing. Our proof of Theorem 2 is via a combinatorial reduction to *bidirected flow* [9]. This reduction also yields a polyhedral characterization of the feasible solutions for all-ror instances.

The methodology behind Theorems 1(b) and 2 is general enough that analogous results can be obtained for covering problems, although we omit the details from this paper. We remark that our application of iterated rounding is somewhat simpler than previous ones since we do not need to "uncross" LP solutions.

Related Work. WMCFT appears in the literature under a variety of names including *cross-free-cut matching* [15] in the unit-capacity case and *packing of a laminar family* [8]. One generalization is the *demand* version [7] in which each commodity i is given a requirement r_i and we require $y_i \in \{0, r_i\}$ for each feasible solution.

The word *bidirected* has two meanings in the literature. We discuss *bidirected flows* later. In contrast, a *bidirected tree* is obtained from an undirected tree by replacing each edge by two antiparallel directed edges. WMCF extends naturally to directed graphs (with directed demand edges). A slight modification of Garg et al.'s hardness proof shows that WMCF on bidirected trees is APX-hard even for unit capacities and weights. On the other hand, WMCF on bidirected trees admits a $(\frac{5}{3} + \epsilon)$-approximation for unit capacities [10] and a 4-approximation [7] for general capacities. On bidirected trees obtained from paths, stars, and spiders the problem can be solved in polynomial time, e.g., by reduction to a max-weight circulation problem, see also [11]. We remark that analogues of Theorems 1(b) and 2 can easily be obtained for bidirected trees.

The WMCF problem on unit capacity trees is equivalent to the *weighted edge-disjoint paths* (WEDP) problem. The polylog-hardness results on general graphs [1, 2] apply to WEDP even with unit weights, however for fixed k, WEDP with at most k commodities is polynomial-time solvable. See e.g. [23, §70.5] for further discussion.

The extreme points of the natural LP for WMCFT arise frequently in the literature of LP-based network design [7, 8, 10, 13, 14, 16, 18, 20, 25]. From this perspective, WMCFT is a natural starting point for an investigation of how large capacities/requirements affect the difficulty of weighted network design problems. As far as we are aware, our results for general weights are novel, but for unit weights Gabow et al. [13, 14] gave a result analogous to Theorem 1(b) for the k-edge-connected spanning subgraph problem, obtaining a $1 + \Theta(1/k)$ threshold of approximation.

Organization of the Paper. Section 1.1 contains some basic definitions and notation. In Section 2 we give the proof of Theorem 1. In Section 3 we provide a proof of Theorem 2. In Section 3.1, we state our polyhedral results. Finally, we suggest some directions for future work in Section 4.

1.1 Formulation

Since the supply graph and demand edges are undirected, without loss of generality we define the commodities by a set of *demand edges* $D = \{\{s_1, t_1\}, \ldots, \{s_k, t_k\}\}$ on vertex set V with a weight w_d assigned to each demand edge $d \in D$. Since we discuss WMCF only on undirected trees, each commodity has a unique path along which flow is sent. For each demand edge d, let its *demand path* p_d be the unique path in G joining the endpoints of d. We thus may represent a multicommodity flow by a vector $\{y_d\}_{d \in D}$ where y_d is the amount of commodity d that is routed (along p_d). Then a flow y is *feasible* if it meets the two conditions

$$\forall d \in D : \qquad y_d \geq 0 \tag{1}$$

$$\forall e \in E : \qquad \sum_{d : e \in p_d} y_d \leq c_e \tag{2}$$

The objective of WMCFT is to find a feasible *integral* y that maximizes $w \cdot y$.

2 Improved Approximation

In this section we obtain a $\min\{3, 1 + 4/\mu + 6/(\mu^2 - \mu)\}$-approximation algorithm for WMCFT, assuming $c_e \geq \mu \geq 2$ for each edge e. The algorithm uses the iterated rounding paradigm, which was used first by Jain [18] and more recently by Singh & Lau [25] and others [4, 13, 20] for network design. The idea is that in each iteration, we use the optimal fractional solution y^* of the natural LP to develop an integral solution. If some demand edge d has value 1 or greater in y^*, we route the integer part and decrease the capacity of edges on p_d accordingly. If $\mathbf{0} < y < \mathbf{1}$ we perform a *relaxation* step; a counting argument (Section 2.3) guarantees that a particular relaxation can always be performed. At the end, we obtain an integral solution which exceeds the capacity of each edge by at most 2, and which has weight at least as large as that of an optimal feasible solution. In Sections 2.1 and 2.2 we show how to compute high-weight feasible solutions from this +2-violating solution.

The natural LP for WMCFT, which we denote by $\mathcal{P}_0$, is as follows:

$$(\mathcal{P}_0) : \qquad \text{maximize } w \cdot y \text{ subject to (1) and (2).}$$

This program has a linear number of variables and constraints, and thus can be solved in polynomial time. Any integral vector y is feasible for $\mathcal{P}_0$ iff it is feasible for the WMCFT instance. However, the linear program has fractional extreme points in general, and thus solving the LP does not give us the type of solution we seek. Nonetheless, optimal solutions to the LP have certain properties that permit an iterated rounding approach, such as the following.

Lemma 3 *Let y^* be an optimal solution to $\mathcal{P}_0$, define $OPT = w \cdot y^*$, and suppose $y^*_d \geq t$ for some $d \in D$ and some integer $t \geq 1$. Reduce the capacity of each edge $e \in p_d$ by t and let OPT' denote the new optimal value of $\mathcal{P}_0$. Then $OPT' = OPT - tw_d$.*

Proof. Let z denote the vector such that $z_d = t$ and $z_{d'} = 0$ for each $d' \neq d$. Then it is easy to see that $y^* - z$ is feasible for the new LP, and hence $OPT' \geq w \cdot (y^* - z) = OPT - tw_d$. On the other hand, where y' denotes the optimal solution to the new LP, it is easy to see that $y' + z$ is feasible for the original LP; so $OPT \geq OPT' + tw_d$. Combining these inequalities, we are done. □

From now on, let OPT denote the optimal value to $\mathcal{P}_0$.

In general terms, our iterated rounding approach works on the following principles. Define the following restricted version of $\mathcal{P}_0$:

$$(\overline{\mathcal{P}_0}): \qquad \text{maximize } w \cdot y \text{ subject to (1) and (2) and } \forall d \in D : y_d \leq 1.$$

Assume for the moment that $\overline{\mathcal{P}_0}$ also has optimal value OPT. We iteratively build an integral solution to $\overline{\mathcal{P}_0}$ with value at least equal to OPT. The first step in each iteration is to solve $\overline{\mathcal{P}_0}$, obtaining solution y^*. If $y^*_d = 0$ for some demand edge d, then we can discard d without affecting the optimal value of $\overline{\mathcal{P}_0}$. If $y^*_d = 1$ for some d, then we can route one unit of flow along p_d and update capacities accordingly. Similar to Lemma 3, the optimal LP value will drop by an amount equal to the weight of the flow that was routed. If neither of these cases applies, we use the following lemma, whose proof appears in Section 2.3.

Lemma 4 *Suppose that y^* is an extreme point solution to $\mathcal{P}_0$, and that $0 < y^*_d < 1$ for each demand edge $d \in D$. Then there is an edge $e \in E$ so that $|\{d \in D : e \in p_d\}| \leq 3$.*

Our algorithm discards the capacity constraint (2) for e from our LP. We call this *contracting* e because the effect is the same as if we had merged the two endpoints of e in the tree G. Pseudocode for our algorithm is given in Figure 1.

IteratedSolver

1. Set $\widehat{y} = \mathbf{0}$
2. If $D = \emptyset$ terminate and return $\widehat{y}$
3. Let y^* be an optimal extreme point solution to $\overline{\mathcal{P}_0}$
4. For each d such that $y^*_d = 0$, discard d
5. For each d such that $y^*_d = 1$, increase $\widehat{y}_d$ by 1, decrease c_e by 1 for each $e \in p_d$, and discard d
6. If neither step 4 nor 5 applied, find e as specified by Lemma 4 and contract e
7. Go to step 2

Fig. 1. The iterated rounding algorithm.

To justify our assumption that $\mathcal{P}_0$ and $\overline{\mathcal{P}_0}$ have the same optimal value, we preprocess the problem as follows. First, we compute any optimal solution y^* to $\mathcal{P}_0$. Then we route the integer part $\lfloor y^* \rfloor$ of the solution (i.e., we initialize $\widehat{y} = \lfloor y^* \rfloor$) and reduce each capacity c_e by $\sum_{d:e \in p_d} \lfloor y^*_d \rfloor$. The residual problem has $\langle y^* \rangle := y^* - \lfloor y^* \rfloor$ as an optimal solution, and since $0 \leq \langle y^* \rangle \leq 1$, our assumption is now justified.

Assuming Lemma 4, we now prove the main properties of our iterated rounding algorithm: it runs in polynomial time, it exceeds each capacity by at most 2, and it produces a solution of value at least OPT.

Property 1 IteratedSolver *runs in polynomial time.*

Proof. Recall that $\mathcal{P}_0$ and $\overline{\mathcal{P}_0}$ can be solved in polynomial time. In each iteration we decrease $|D| + |E|$, so polynomially many iterations occur, and the result follows. □

Property 2 *The integral solution computed by* IteratedSolver *violates each capacity constraint* (2) *by at most* 2.

Proof. Consider what happens to any given edge e during the execution of the algorithm. In the preprocessing and in each iteration, the flow routed through e equals the decrease in its residual capacity. If in some iteration, e's residual capacity is decreased to 0, all demand paths through e will be discarded in the following iteration. Thus if e is not contracted, its capacity constraint (2) will be satisfied by the final solution.

The other case is that we contract e in step 6 of some iteration because e lies on at most 3 demand paths. The residual capacity of e is at least 1, and at most one unit of flow will be routed along each of these 3 demand paths in future iterations. Hence the final solution violates (2) for e by at most $+2$. □

Property 3 *The integral solution computed by* ITERATEDSOLVER *has objective value at least equal to OPT.*

Proof. When we contract an edge e we just remove a constraint from $\overline{\mathcal{P}_0}$, which cannot decrease the optimal value of $\overline{\mathcal{P}_0}$ since it is a maximization LP. In every other iteration and in preprocessing, Lemma 3 implies that the LP optimal value drops by an amount equal to the increase in $w \cdot \widehat{y}$. When termination occurs, the optimal value of $\overline{\mathcal{P}_0}$ is 0. Thus the overall weight of flow routed must be at least as large as the initial value of OPT. □

2.1 Minimum Capacity $\mu = 2$

As per Property 2, our iterated solver may exceed some of the edge capacities. When $c_e \geq \mu \geq 2$ for each edge e we can invoke the following theorem, which appears as Thm. 6 in [8], to produce a high-weight feasible solution.

Theorem 5 (Cheriyan, Jordán, Ravi) *Suppose that $\widehat{y}$ is a nonnegative integral vector so that for each edge e, the constraint* (2) *is violated by at most a* multiplicative *factor of* 2 *by $\widehat{y}$. Then in polynomial time, we can find an integral vector y' with $w \cdot y' \geq (w \cdot \widehat{y})/3$, and $0 \leq y' \leq \widehat{y}$, and such that y' satisfies all constraints* (2)*.*

The algorithm as literally described in [8] is actually pseudo-polynomial, but it is straightforward to modify it to have polynomial running time — see Appendix B. We are now able to prove part (a) of Theorem 1.

Proof (of Theorem 1(a)). Let $\widehat{y}$ be the output of ITERATEDSOLVER. Since $c_e \geq 2$ for each edge e, and since by Property 2 each edge's capacity is additively violated by at most $+2$, Theorem 5 applies. Thus y' is a feasible solution to the WMCFT instance with objective value $w \cdot y' \geq w \cdot \widehat{y}/3 \geq OPT/3$, using Property 3. Finally, since $\mathcal{P}_0$ is an LP-relaxation of the WMCFT problem, OPT is at least equal to the optimal WMCFT value, and so y' is a 3-approximate feasible integral solution. □

2.2 Arbitrary Minimum Capacity

Given the infeasible solution $\widehat{y}$ produced by ITERATEDSOLVER, we want to reduce $\widehat{y}$ in a minimum-weight way so as to attain feasibility. For each edge e let $f_e = \max\{0, \sum_{d:e \in p_d} \widehat{y}_d - c_e\}$, i.e. f_e is the amount by which $\widehat{y}$ violates the capacity of e. Note now that a reduction z with $0 \leq z \leq \widehat{y}$ makes $\widehat{y} - z$ a feasible (integral) WMCFT solution if and only if z is a feasible (integral) solution to the following linear program.

$$(\mathcal{P}_c): \qquad \text{minimize } w \cdot z \text{ subject to } 0 \leq z \leq \widehat{y} \text{ and } \forall e \in E: \sum_{d:e \in p_d} z_d \geq f_e.$$

Notice that $\mathcal{P}_c$ is a covering analogue of $\mathcal{P}_0$ (with added upper bounds). Furthermore, the integer program $\mathcal{P}_c$ can be 2-approximately solved using Jain's iterated rounding framework [18].

Theorem 6 *There is a polynomial-time algorithm which returns an integral feasible solution $\widehat{z}$ for $\mathcal{P}_c$ such that $w \cdot \widehat{z}$ is at most twice the LP optimal value of $\mathcal{P}_c$.*

Proof. The idea is very similar to the main result of [18] but simpler in that no *uncrossing* is needed, because we already have a tree structure. Hence we only sketch the details. In each iteration, we obtain an extreme point optimal solution z^* to the linear program $\mathcal{P}_c$. We increase $\widehat{z}$ by the integer part of z^* and accordingly decrease the requirements f. If $z^*_d = 0$, d is discarded. Finally if $0 < z^* < 1$ a token redistribution argument of Jain shows that some $d^* \in D$ has $z^*_{d^*} \geq 1/2$. In this case we increase $\widehat{z}_{d^*}$ by 1 and update the requirements accordingly. Standard arguments then give the claimed bound on the cost of $\widehat{z}$ and polynomial running time. □

Here is how we use Theorem 6 to approximate WMCFT instances on trees.

Proof (of Theorem 1(b)). Notice that $z = \frac{2}{\mu+2}\widehat{y}$ is a feasible fractional solution to $\mathcal{P}_c$. Hence, the optimal value of $\mathcal{P}_c$ is at most $\frac{2}{\mu+2}\widehat{y} \cdot w$. Thus the solution $\widehat{z}$ produced by Theorem 6 satisfies $\widehat{z} \cdot w \leq \frac{4}{\mu+2}\widehat{y} \cdot w$, so $\widehat{y} - \widehat{z}$ is a feasible solution to the WMCFT problem, with $w \cdot (\widehat{y} - \widehat{z}) \geq (1 - \frac{4}{\mu+2})\widehat{y} \cdot w \geq (1 - \frac{4}{\mu+2})OPT$. This gives us a $1/(1 - \frac{4}{\mu+2}) = 1 + 4/\mu + O(1/\mu^2)$ approximation algorithm for WMCFT.

To obtain the exact bound claimed in Theorem 1(b), we refine this slightly by taking a two-round approach. In the first round we set f_e to be the characteristic vector of those edges which $\widehat{y}$ violates by $+2$, obtaining $\widehat{y}' := \widehat{y} - \widehat{z}$. Then $\widehat{y}'$ has only $+1$ additive violation, and the same reasoning as before shows $\widehat{y}' \cdot w \geq (1 - \frac{2}{\mu+2})OPT$. The second round analogously extracts from $\widehat{y}'$ a feasible solution with weight at least $(1 - \frac{2}{\mu+1})\widehat{y}' \cdot w$. This gives approximation ratio $1/(1 - \frac{2}{\mu+2})(1 - \frac{2}{\mu+1}) = 1 + 4/\mu + 6/(\mu^2 - \mu)$, as desired. □

2.3 Proof of Lemma 4

First, we need the following simple counting argument.

Lemma 7 *Let T be a tree with n vertices and let n_i denote the number of its vertices that have degree i. Then $n_1 > (n - n_2)/2$.*

Proof. Using the handshake lemma and the fact that T has $n-1$ edges, we have $2(n-1) = \sum_i i \cdot n_i$. But $\sum_i i \cdot n_i \geq n_1 + 2n_2 + 3(n - n_1 - n_2) = 3n - 2n_1 - n_2$ and hence $2n - 2 \geq 3n - 2n_1 - n_2$. Solving for n_1 gives $n_1 \geq (n - n_2 + 2)/2$ as needed. □

Proof (of Lemma 4). Using basic facts from polyhedral combinatorics, it follows that there exists a set $E^* \subset E$ of edges with $|E^*| = |D|$ such that y^* is the unique solution to

$$\sum_{d \in D : e \in p_d} y_d = c_e \qquad \forall e \in E^*. \tag{3}$$

In particular, the characteristic vectors of the sets $\{d : e \in p_d\}$ for $e \in E^*$ are linearly independent.

Contract each edge of $E \backslash E^*$ in (V, E), resulting in the tree $T' = (V', E^*)$; call elements of V' *nodes*. We now use a counting argument to establish the existence of the desired edge e within E^*.

We call the two ends of each $d \in D$ *endpoints* and say that node $v' \in V'$ *owns* k endpoints when the degree of v' in (V', D) is k.

First, consider any node $v' \in V'$ that has degree 2 in T'; let e_1, e_2 be its incident edges in T'. If v' owns no endpoints then $\{d : e_1 \in p_d\} = \{d : e_2 \in p_d\}$, contradicting linear independence. If v' owns exactly one endpoint, the symmetric difference $\{d : e_1 \in p_d\} \triangle \{d : e_2 \in p_d\}$ consists of a single demand edge; but since y^* satisfies (3), $\mathbf{0} < y^* < \mathbf{1}$, and c is integral, this is a contradiction. Hence v' owns two or more endpoints.

If there exists a leaf node v' of T' that owns at most 3 endpoints then we are done, since this implies that the edge of E^* incident to v', viewed in the original graph, lies on a most 3 demand paths. Otherwise, we apply a counting argument to T', seeking a contradiction. Let n_i denote the number of nodes of T' of degree i. Then our previous arguments establish that the total number of endpoints is at least $4n_1 + 2n_2$. Lemma 7 then shows that the total number of endpoints is more than $2(|V'| - n_2) + 2n_2 = 2|V'| > 2|E^*| = 2|D|$. This is the desired contradiction, since there are only $2|D|$ endpoints in total. □

We remark that Lemma 4 is tight in the following sense: if we replace the bound $|\{d \in D : e \in p_d\}| \leq 3$ with $|\{d \in D : e \in p_d\}| \leq 2$, the resulting statement is false. An example of an extreme point solution for which the modified version fails, due to Cheriyan et al. [8], is given in Figure 2.

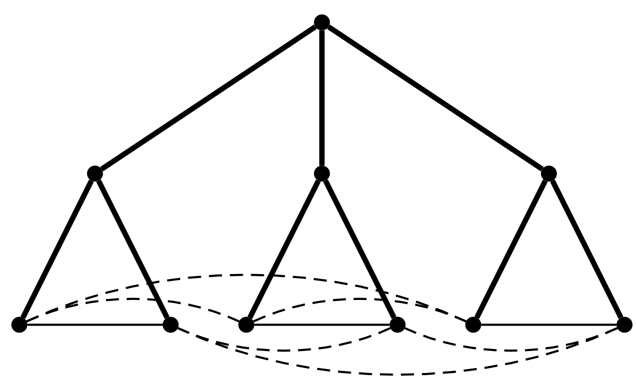

Fig. 2. An extreme point solution to $\mathcal{P}_0$. There are 9 edges in the supply graph, shown as thick lines; each has capacity 1. There are 9 demand edges, shown as thin lines; the solid ones have value 1/2, and the dashed ones have value 1/4. This is a tight example for Lemma 4 because each edge lies on at least three demand paths.

3 Exact Solution for Spiders

In this section we show that WMCFT can be exactly solved in polynomial time when the supply graph is a spider. (A *spider* is a tree with exactly one vertex of degree greater than 2.) Call the vertex of degree ≥ 3 the *root* of the spider. Call each maximal path having the root as an endpoint a *leg* of the spider. Observe that in WMCFT when (V, E) is a spider, each demand path p_d either goes through the root, or else lies within a single leg.

Definition 8 *Consider an instance of WMCFT on graph (V, E). With respect to a chosen root vertex $r \in V$, a demand edge d is said to be*

root-using, *if r is an internal vertex of p_d;*

radial*, if one endpoint of d is a descendant of the other, with respect to the orientation of (V,E) induced by the root r.*

The instance is all-ror *(short for "all root-using or radial") if there exists a choice of $r \in V$ for which every demand edge is either root-using or radial.*

Instances with only radial demand paths can be exactly solved via the LP $\mathcal{P}_0$ since the constraint matrix is unimodular. Instances with only root-using demand paths can be solved using a matching approach, see e.g. the work of Nguyen [21]. To solve all-ror instances in general we use *bidirected flows*, which were introduced by Edmonds and Johnson [9]. Bidirected flow problems can be solved via a combinatorial reduction to b-matching (e.g., see [23]) which increases the instance size by a constant factor. We present in this section a reduction from all-ror WMCFT to bidirected flow.

A bidirected graph is an undirected graph together with, for each edge e and each of its endpoints $v \in e$, a sign $\sigma_{v,e} \in \{-1, +1\}$. Thus an edge can have two negative ends, two positive ends, or one end of each type; these are respectively called *negative* edges, *positive* edges, and *directed* edges. We will speak of directed edges as having the $+1$ end as their head and -1 end as their tail. An instance of capacitated max-weight bidirected flow is an integer program of the following form.

$$\text{maximize} \sum_{e \in E} \pi_e x_e \tag{4}$$

$$\forall v \in V: \quad a_v \leq \sum_{e \ni v} \sigma_{v,e} x_e \leq b_v \tag{5}$$

$$\forall e \in E: \quad \ell_e \leq x_e \leq u_e \tag{6}$$

$$x \text{ integral} \tag{7}$$

When $a = b = \mathbf{0}$ and all edges are directed, (4)–(7) becomes a max-weight circulation problem; when all edges are positive and $a = \mathbf{0}$, (4)–(7) becomes a b-matching problem. We now describe the reduction.

Proof (of Theorem 2). Let r denote the root vertex, i.e., assume every demand edge is either radial or root-using with respect to r. We construct a bidirected graph whose underlying undirected graph is $(V, E \cup D)$. Make each edge $e \in E$ directed, with head pointing towards r in the tree (V,E). We make each root-using $d \in D$ a positive edge; we make each radial $d \in D$ a directed edge, with head pointing away from r. See Figure 3 for an illustration.

Set $a_r = -\infty$, $b_r = +\infty$ and $a_v = b_v = 0$ for each $v \in V \backslash \{r\}$ in the bidirected flow problem (4)–(7). For each demand edge $d \in D$, call $C(d) := \{d\} \cup p_d$ the *demand cycle* of d. For a set F let χ^F denote the characteristic vector of F. Our choices of signs for the endpoints ensure that for each demand cycle C, its characteristic vector $x = \chi^{C(d)}$ satisfies constraint (5). Moreover, any linear combination of these vectors is easily seen to satisfy (5), and the following converse holds.

Claim 9 *Any x satisfying* (5) *is a linear combination of characteristic vectors of demand cycles.*

We defer the proof of Claim 9 to Appendix C.

By Claim 9, we may change the variables in the optimization problem from x to instead have one variable y_d for each $d \in D$; the variables are thus related by $x = \sum_d y_d \chi^{C(d)}$. In the bidirected optimization problem, set $\ell_e = 0, u_e = c_e$ for each $e \in E$, and $\ell_d = 0, u_d = +\infty$ for each $d \in D$. Rewriting (6) in terms of the new variables gives precisely the constraints (1) and (2). In other words, feasible integral flows x correspond bijectively to feasible integral solutions y for the WMCFT

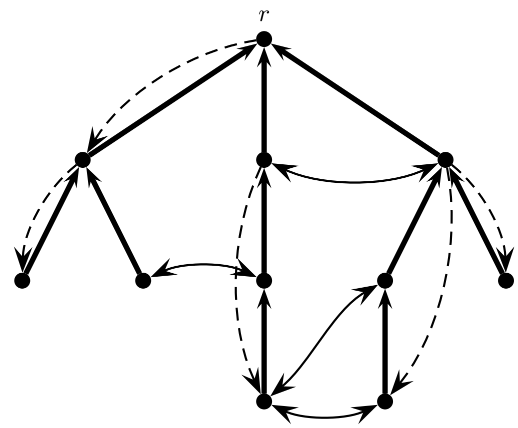

Fig. 3. An all-ror multicommodity flow instance. The tree graph (V, E) is depicted using thick lines, and the demand edges D are thin. Radial demand edges are dashed and root-using demand edges are solid. The root is r. An arrowhead denotes a positive endpoint, while the remaining endpoints are negative; these signs correspond to the reduction in the proof of Theorem 2.

instance. Setting $\pi_d = w_d$ for $d \in D$ and $\pi_e = 0$ for $e \in E$, the objective function of (4) represents the weight for y, completing the reduction.

As mentioned earlier, this bidirected flow problem can in turn be reduced to a b-matching problem with a constant factor increase in the size of the problem. Using the strongly polynomial b-matching algorithm of Anstee [3], the proof of Theorem 2 is complete. □

3.1 Polyhedral Results

The reduction used in the proof of Theorem 2 can also be used to derive the following result.

Theorem 10 *The convex hull of all integral feasible solutions in an all-ror WMCFT problem has the following description:*

$$y_d \geq 0, \qquad \forall d \in D \tag{8}$$

$$\sum_{e \in p_d} y_d \leq c_e, \qquad \forall e \in E \tag{9}$$

$$\sum_{d \in D} y_d \lfloor |p_d \cap F|/2 \rfloor \leq \lfloor c(F)/2 \rfloor, \qquad \forall F \subset E \tag{10}$$

We defer the proof of Theorem 10 to Appendix D. Interestingly, results of Garg et al. [15, preliminary version] show that (8)–(10) is also integral in *unit-capacity* WMCFT. It is possible to synthesize our Theorems 2 and 10 with corresponding results of [15] for the unit-capacity case, by "gluing" all-ror instances at capacity-1 edges, but we omit these specialized results from the current paper.

Caprara and Fischetti [6] gave a strongly polynomial-time algorithm to separate over the family (10) of inequalities. The ubiquity of the polyhedral formulation (8)–(10) suggests that it might be useful in designing a better approximation algorithm for WMCFT. One roadblock is that we do not know how to perform *uncrossing* (see, e.g., [18]) for that LP.

4 Closing Remarks

For the problem of finding a k-edge connected subgraph with the smallest number of edges, it is known [13, 14] that the best approximation ratio possible (if $\mathsf{P} \neq \mathsf{NP}$) is $1+\Theta(1/k)$. We have proven a similar phenomenon for WMCFT (in terms of μ) and these results rely on similar techniques, notably iterated LP rounding. It would be nice to resolve the following outstanding question: for the problem of finding a *min-weight* k-edge connected subgraph, does the best possible approximation ratio decrease as k increases?

There is a close relation between WMCFT and its "demand" version where every flow variable is restricted according to $y_d \in \{0, r_d\}$ for some constants $\{r_d\}_{d \in D}$. E.g., combining IteratedSolver and Cor. 3.5 of [7], we obtain a $1 + O(1/\sqrt{\mu})$ approximation for demand WMCFT where μ is the ratio of the minimum capacity to the maximum demand. An interesting open question is whether the $O(d_{\max})$-relaxed integrality gap of the demand analogue of $\mathcal{P}_0$ is 1, where $d_{\max}$ is the maximum demand; we have shown that this holds for the unit-demand case, while Shepherd and Vetta [24] showed it holds for general demands when G is a star. To prove that it holds for general demands on a tree, or even general demands on a line, seems to require new ideas.

References

1. M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. In *Proc. 46th FOCS*, pages 226–244, 2005.
2. M. Andrews and L. Zhang. Logarithmic hardness of the undirected edge-disjoint paths problem. *J. ACM*, 53(5):745–761, 2006. Preliminary version appeared in *Proc. 37th STOC*, pages 276–283, 2005.
3. R. P. Anstee. A polynomial algorithm for b-matchings: An alternative approach. *Inf. Process. Lett.*, 24(3):153–157, 1987.
4. N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree bounded directed network design. Technical Report RC24347, IBM, 2007.
5. G. Calinescu, A. Chakrabarti, H. J. Karloff, and Y. Rabani. Improved approximation algorithms for resource allocation. In *Proc. 9th IPCO*, pages 401–414, 2002.
6. A. Caprara and M. Fischetti. $\{0, \frac{1}{2}\}$-Chvátal–Gomory cuts. *Math. Program.*, 74(3):221–235, 1996.
7. C. Chekuri, M. Mydlarz, and F. B. Shepherd. Multicommodity demand flow in a tree and packing integer programs. *ACM Trans. Algorithms*, 3(3):27, 2007. Extended abstract appeared in *Proc. 30th ICALP*, pages 410–425, 2003.
8. J. Cheriyan, T. Jordán, and R. Ravi. On 2-coverings and 2-packings of laminar families. In *European Symposium on Algorithms*, pages 510–520, 1999.
9. J. Edmonds and E. Johnson. Matching: A well-solved class of integer linear programs. In R. Guy, H. Hanani, N. Sauer, and J. Schonheim, editors, *Proceedings, Calgary International Conference on Combinatorial Structures and their Applications*, pages 82–92. Gordon and Breach, 1970.
10. T. Erlebach and K. Jansen. Conversion of coloring algorithms into maximum weight independent set algorithms. In J. D. P. Rolim et al., editor, *Proc. ICALP Satellite Workshops*, pages 135–146, 2000.
11. T. Erlebach and K. Jansen. The maximum edge-disjoint paths problem in bidirected trees. *SIAM J. Discrete Math.*, 14(3):326–355, 2001. Preliminary version appeared as "Maximizing the Number of Connections in Optical Tree Networks" in *Proc. 9th ISAAC*, pages 179–188, 1998.
12. S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976. Preliminary version appeared in *Proc. 16th FOCS*, pages 184–193, 1975.
13. H. N. Gabow and S. Gallagher. Iterated rounding algorithms for the smallest k-edge-connected spanning subgraph. In *Proc. 19th SODA*, pages 550–559, 2008.
14. H. N. Gabow, M. X. Goemans, Éva Tardos, and D. P. Williamson. Approximating the smallest k-edge connected spanning subgraph by LP-rounding. In *Proc. 16th SODA*, pages 562–571, 2005.
15. N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, May 1997. Preliminary version appeared in *Proc. 20th ICALP*, pages 64–75, 1993.

16. M. X. Goemans. Minimum bounded degree spanning trees. In *Proc. 47th FOCS*, pages 273–282, 2006.
17. I. B.-A. Hartman. Optimal k-colouring and k-nesting of intervals. In *Proc. 4th Israel Symposium on Theory of Computing and Systems*, pages 207–220, 1992.
18. K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001. Preliminary version appeared in *Proc. 39th FOCS*, pages 448–457, 1998.
19. V. Kann. *On the Approximability of NP-complete Optimization Problems*. PhD thesis, Royal Institute of Technology Stockholm, 1992.
20. L. C. Lau, J. S. Naor, M. R. Salavatipour, and M. Singh. Survivable network design with degree or order constraints. In *Proc. 39th STOC*, pages 651–660, 2007.
21. T. Nguyen. On the disjoint paths problem. *Oper. Res. Lett.*, 35(1):10–16, 2007.
22. P. Raghavan and C. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.
23. A. Schrijver. *Combinatorial optimization*. Springer, New York, 2003.
24. F. B. Shepherd and A. Vetta. The Demand-Matching Problem. *Mathematics of Operations Research*, 32(3):563–578, 2007. Preliminary version appeared in *Proc. 9th IPCO*, pages 457-474, 2002.
25. M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proc. 39th STOC*, pages 661–670, 2007.

A Hardness of WMCFT with Constant Lower Bounds

Theorem 11 *For some $\epsilon > 0$, for all integers $\mu \geq 2$, it is* NP*-hard to approximate WMCFT to a factor of less than $1+\epsilon/\mu$, even when restricted to instances where all capacities are at least μ, and all weights are unit.*

Proof. We extend Garg et al.'s construction [15, Thm. 4.2]. Their reduction is from *3-bounded maximum three-dimensional matching* (MAX 3DM–3); an instance consists of three disjoint sets X, Y, Z of size n and a family of triples $S \subset X \times Y \times Z$ such that each element is in at most 3 triples. The objective is to find a maximum-size disjoint set of triples from S. If $|S| < n$ elements appearing in no triples can be discarded, so WOLOG $|S| \geq n$.

Kann [19] showed that MAX 3DM–3 is MAX SNP-complete, hence by the PCP theorem, for some $\delta > 0$ it is NP-hard to approximate it within a ratio of $1 + \delta$. A greedy argument easily shows the optimal value of MAX 3DM–3 is always at least $|S|/7$. Hence it is NP-hard to additively approximate MAX 3DM–3 within $|S|\delta/7$.

The construction given in [15] is an instance of unit-weight WMCFT on a tree which has optimal value $t + |S|$ where t is the maximum number of disjoint triples in S. Let (V, E) denote the tree obtained from this construction; it satisfies $|E| = 3(|S|+n)$. To obtain a lower bound μ on capacity, we perform the following for each edge $uv \in E$: (1) add a new leaf u' and a new edge uu' to the tree; (2) add a new unit-weight demand edge $u'v$ to D; (3) increase the capacity of uv by μ and set the capacity of uu' to be μ. We illustrate the overall modification in Figure 4.

An easy alteration argument shows that there exists an optimal solution y to the modified WMCFT problem such that $y_d = \mu$ for each new demand edge d. It then follows that its optimal value is $t+|S|+\mu|E|$. Furthermore, since $t+|S|+\mu|E| \leq (6\mu+2)|S|$, approximating the WMCFT instance to a ratio less than $1 + \frac{|S|\delta/7}{(6\mu+2)|S|} = 1 + \Theta(1/\mu)$ is NP-hard. □

B Polynomial Running Time of Packing Conversion

Cheriyan et al.'s algorithm applies to a multiset $\mathcal{P}$ of demand edges, and has running time polynomial in $|\mathcal{P}|$. Directly applying their algorithm to the multiset containing y_d copies of each demand

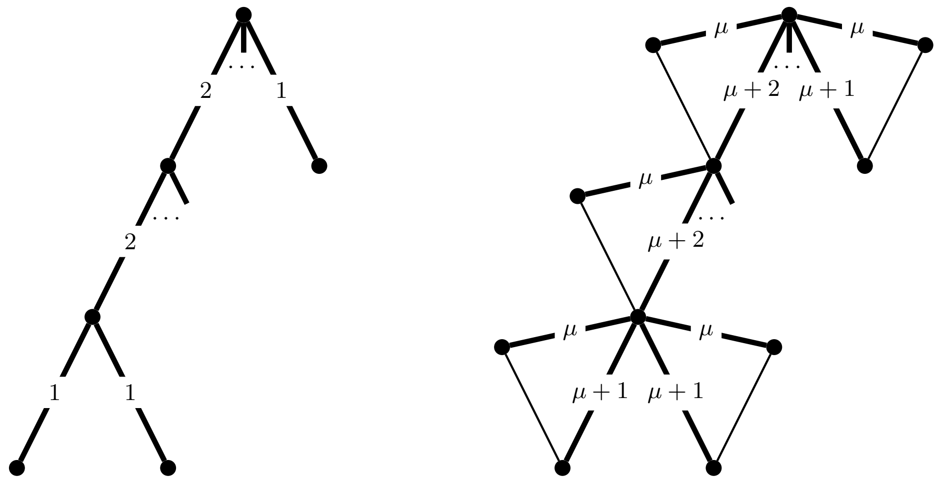

Fig. 4. Left: a portion of the capacitated tree used in the hardness construction of [15]. Demand edges are not shown. Right: the modified capacitated tree (thick) used in the proof of Theorem 11, and the new demand edges (thin). The old demand edges are still not shown, but present.

edge d would give us a correct answer y', but the running time would be pseudo-polynomial in the length of the WMCFT instance's description, i.e., it would run in $\mathrm{poly}(\sum_d y_d) = \mathrm{poly}(|E| \max_e c_e)$ time, whereas we can only allow polylogarithmic dependence on $\max_e c_e$.

To modify this for our purposes, define $\widetilde{y}_d = \lfloor y_d/2 \rfloor$ for each demand edge d and let $\overline{y} = y - 2\widetilde{y}$, so $\overline{y}$ is a 0-1 vector. Define, for each edge e, $c'_e := c_e - \sum_{d:e\in p_d} \widetilde{y}_d$ and notice that $\overline{y}$ violates the capacities c' by at most a factor of 2. Apply the algorithm of Theorem 5 to $\overline{y}$, producing a solution $\overline{y}'$ which is feasible for c' with $w\cdot\overline{y}' \geq w\cdot\overline{y}/3$. This application is polynomial-time since $\sum_d \overline{y}_d \leq |D|$. Define $y' = \overline{y}' + \widetilde{y}$; then it is immediate that y' is feasible for capacities c. Furthermore,

$$w \cdot y' = w \cdot (\overline{y}' + \widetilde{y}) \geq \frac{1}{3} w \cdot \overline{y} + w \cdot \widetilde{y} \geq \frac{1}{3} w \cdot (\overline{y} + 2\widetilde{y}) = (w \cdot y)/3,$$

as needed.

C Proof of Claim 9

Let $x' = x - \sum_d x_d \chi^{C(d)}$, and observe that x' also satisfies (5). Moreover, as each particular demand edge d^* occurs only in one demand cycle, namely $C(d^*)$, we have $x'_{d^*} = x_{d^*} - \sum_d x_d \chi^{C(d)}_{d^*} = x_{d^*} - x_{d^*} = 0$ for each $d^* \in D$. In other words, x' vanishes on D.

Now consider any leaf $v \neq r$ of G and its incident edge $uv \in E$. Since x' satisfies (5) at v and x' is zero on every edge incident to v except possibly uv, we deduce that $x_{uv} = 0$. By induction we can repeat this argument to show that x' also vanishes on all of E, so $x' = 0$. Then $x = x' + \sum_d x_d \chi^{C(d)} = \sum_d x_d \chi^{C(d)}$, which proves Claim 9.

D Proof of Theorem 10

It is obvious that constraints (8) and (9) are valid. To see that the constraint (10) is valid, notice that it can be obtained as a Chvátal-Gomory cut: give coefficient 1/2 to each constraint (9) for

$e \in F$. This establishes necessity, and the rest of the proof will establish sufficiency of the constraints (8)–(10).

Our starting point is the following polyhedral characterization, which appears as Cor. 36.3a in Schrijver [23], and deals with the special case of bidirected flow when $a = b$ and $\ell = \mathbf{0}$.

Proposition 12 *Let σ denote the signs of a bidirected graph. The convex hull of the integer solutions to*

$$\forall e \in E : 0 \le x_e \le u_e \qquad \forall v \in V : \sum_{e \ni v} \sigma_{v,e} x_e = b_v \tag{11}$$

(i.e. the convex hull of all feasible integral bidirected flows) is determined by Equation (11) *together with the constraints*

$$x(\delta(U) \backslash F) - x(F) \ge 1 - u(F) \tag{12}$$

where $U \subseteq V$ and $F \subseteq \delta(U)$ with $b(U) + u(F)$ odd.

In order to apply Proposition 12 to the construction in the proof of Theorem 2, we set $a_r = b_r = 0$ and add a loop at r with both of its endpoints negative. Further, we change the definition of $C(d)$ to include this loop whenever d is a radial demand edge; then it is not hard to show that, just as before, feasible bidirected flows x correspond bijectively to feasible multicommodity flows y.

Now apply Proposition 12 to the construction. Recall that the edge set of the bidirected graph is $D \cup E$. Since $u_d = +\infty$ for $d \in D$, the constraint (12) is vacuously true unless $F \subset E$. Furthermore, recall that b is the all-zero vector and $u_e = c_e$ for $e \in E$. Rearranging, we obtain the following description of the convex hull of all integral feasible bidirected flows:

$$\forall e \in E : 0 \le x_e \le c_e \qquad \forall d \in D : 0 \le x_d \qquad \forall v \in V : \sum_{e \ni v} \sigma_{v,e} x_e = 0 \qquad \text{and}$$

$$x(F) - x(\delta(U))/2 \le (c(F) - 1)/2 \quad \text{for } U \subseteq V, F \subseteq E \cap \delta(U), c(F) \text{ odd}$$

Rewriting in terms of the y variables, and collecting like terms, yields

$$\forall d \in D : 0 \le y_d \qquad \forall e \in E : \sum_{e \in p_d} y_d \le c_e \qquad \text{and} \tag{13}$$

$$\sum_d y_d(|p_d \cap F| - |C(d) \cap \delta(U)|/2) \le (c(F) - 1)/2 \quad \text{for } U \subseteq V, F \subseteq E \cap \delta(U), c(F) \text{ odd} \tag{14}$$

For any fixed choice of $F \subseteq E$, let $U^*_F \subseteq V$ be the unique set such that $\delta(U^*_F) \cap E = F$ and $r \notin U$. We claim that for this F, constraint (14) is tightest for $U = U^*_F$. To see this, note first that $C(d) \cap \delta(U)$ is always even (since in traversing the cycle $C(d)$, we enter U as many times as we leave); second, that $|C(d) \cap \delta(U^*_F)|/2 = \lceil |p_d \cap F|/2 \rceil$; third, that for any other U' such that $F \subseteq \delta(U')$, $|C(d) \cap \delta(U')|/2$ is an integer greater than or equal to $|p_d \cap F|/2$.

Hence, there is no loss of generality in assuming $U = U^*_F$ in constraint (14). Rewriting, it becomes

$$\sum_d y_d(|p_d \cap F| - \lceil |p_d \cap F|/2 \rceil) \le (c(F) - 1)/2 \quad \text{for } c(F) \text{ odd};$$

finally, since $t = \lfloor t/2 \rfloor + \lceil t/2 \rceil$ for all integers t, the theorem follows.