Multicommodity Flow in Trees: Packing via Covering and Iterated Relaxation*

Jochen Könemann, Ojas Parekh, and David Pritchard

February 7, 2011

Abstract

We consider the max-weight integral multicommodity flow problem in trees. In this problem we are given an edge-, arc-, or vertex-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 is APX-hard and a 4-approximation for the edge- and arc-capacitated versions is known. Some special cases are exactly solvable in polynomial time, including when the graph is a path or a star.

We show that all three versions of this problems fit in a common framework: first, prove a counting lemma in order to use the *iterated LP relaxation* method; second, solve a covering problem to reduce the resulting infeasible solution back to feasibility without losing much weight. The result of the framework is a $1 + O(1/\mu)$-approximation algorithm where μ denotes the minimum capacity, for all three versions. A complementary hardness result shows this is asymptotically best possible. For the covering analogue of multicommodity flow, we also show a $1 + \Theta(1/\mu)$ approximability threshold with a similar framework.

When the tree is a *spider* (i.e. only one vertex has degree greater than 2), we give a polynomial-time exact algorithm and a polyhedral description of the convex hull of all feasible solutions. This holds more generally for instances we call *root-or-radial*.

1 Introduction

In the max-weight integral multicommodity flow problem (WMF), 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 capacities. We distinguish between three versions of the problem: in edge-WMF each edge $e \in E$ has a capacity c_e; in arc-WMF each of the $2|E|$ directed arcs (u, v) with $\{u, v\} \in E$ has a capacity c_{uv}; in vertex-WMF each vertex $v \in V$ has a capacity c_v. We allow $s_i = t_i$ only in the vertex-capacitated setting, representing a path using just one vertex. 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 versions ($k = 1$) of WMF 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 arc- and edge-versions are NP-complete — see Even, Itai, and Shamir [15]. Let $n := |V|$. Results [1, 20] on the *edge-disjoint paths* problem show more strongly that arc-WMF is NP-hard to $|E|^{\frac{1}{2}-\epsilon}$-approximate, and edge-WMF cannot be approximated better than $\log^{\frac{1}{2}-\epsilon}(n)$ unless $\mathsf{NP} \subset \mathsf{ZPTIME}(n^{\mathrm{polylog}(n)})$.

An easier and significant special case of WMF is where the supply graph G is a tree, which we denote by WMFT. Garg, Vazirani and Yannakakis [18] considered the unit-weight case of edge-WMFT 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. Garg et al. show that edge-WMFT 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 same methods show arc-WMFT with unit

*A preliminary version appeared in Proc. 6th WAOA, pages 1–14, 2008.

capacity, and all WMFTs on stars, are in P. The case where G is a path (so-called *interval packing*) is also polynomial-time solvable [5, 8, 21], e.g. by linear programming since the natural LP has a totally unimodular constraint matrix. Arc-WMF on unit-capacity bidirected trees admits a $(\frac{5}{3}+\epsilon)$-approximation algorithm [13]. In general, the best result for edge- and arc-WMFT is a 4-approximation of Chekuri, Mydlarz and Shepherd [8]. Vertex-WMFT has not been explicitly studied as far as we are aware, although we observe that techniques of [8] yield a 5-approximation (see the appendix). Note that edge-WMF can be reduced to vertex-WMF by subdividing each edge, moving that edge's capacity onto the new vertex, and setting all other vertex capacities to be infinite.

Results. Our first important technical contribution is to show that the *iterated relaxation* method [25, 34, 3, 16, 24, 4]) can be applied, yielding an integral solution with optimal value or better but exceeding edge capacities additively — by +2 for edge-WMFT and by +6 for arc-WMFT and vertex-WMFT. A *counting lemma* is what we use to show that iterated relaxation applies. In the words of Chekuri et al. [8] we show for edge-WMFT that the "c-relaxed integrality gap" is 1 for $c = 2$; they conjectured it would hold for some constant c.

We will use μ to denote the minimum capacity in the WMFT instance. When the minimum capacity μ is $\Omega(\log |V|)$, randomized rounding [12, 31] gives a $1 + O(\log |V|/\mu)$-approximation for WMFT. Chekuri et al. [8] anticipated moreover that WMFT becomes easier to approximate as the minimum capacity increases; by plugging our iterated relaxation results into [8, Cor. 3.5] we get a $1 + O(1/\sqrt{\mu})$-approximation. Our first main result improves the best approximation ratio for edge-WMFT when $\mu \geq 2$.

Theorem 1. *For edge-WMFT, 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* $\mu \geq 2$*.*

Theorem 1 starts from the iterated relaxation method. Then, we decrease the additively-violating solution towards feasibility, without losing too much weight. In particular for part (a) we use a theorem of Cheriyan, Jordán and Ravi [9] which basically states that we can reduce the load of any tree flow by a factor of 2, while keeping at least 1/3 of the weight. Part (b) relies on an auxiliary *covering* problem; every feasible cover, when subtracted from the +2-violating edge-WMFT solution, results in a feasible edge-WMFT solution. Jain's *iterated rounding* algorithm [22] gives a 2-approximation for the auxiliary problem. Moreover, we use a crucial property of Jain's algorithm: its cost is even at most twice the best *fractional* solution; call this property *LP-relative*[1]. Together, these pieces give the $1 + O(1/\mu)$-approximation algorithm.

The same framework is flexible enough to work in the settings of arc- or vertex-capacities:

Theorem 2. *For arc-WMFT and vertex-WMFT, there are polynomial-time* $1 + O(1/\mu)$*-approximation algorithms.*

This theorem needs more involved counting lemmas. For the auxiliary covering problem we cannot use Jain's algorithm precisely, nonetheless we can use the counting lemmas a second time to show that iterated rounding gives the needed LP-relative $O(1)$-approximation. Concurrently with preparation of this manuscript, a $\max\{2, 3 - \frac{2}{\mu}\}$-approximation was found for arc-WMFT when all arcs have capacity *exactly* μ [28].

Rearranging our tools somewhat, we show that a similar phenomenon holds for *WMFT-cover*, which is the problem of finding integral s_i-t_i flows of value y_i so as to *minimize* $\sum_i w_i y_i$ and so that the amount of flow through each edge/arc/vertex is at *least* its capacity.

Theorem 3. *For WMFT-cover, with either edge, arc, or vertex capacities, there is a polynomial-time* $1 + O(1/\mu)$*-approximation algorithm.*

It is natural to ask if the above results are best possible, asymptotically with respect to μ. We show that this is indeed the case by extending Garg et. al's proof that edge-WMFT is APX-hard.

Theorem 4. *For some fixed* $\epsilon > 0$*, for all* $\mu \geq 1$*, it is* NP*-hard to approximate WMFT or WMFT-cover with edge, arc, or vertex capacities within ratio* $1 + \epsilon/\mu$*, on instances with minimum capacity* μ*.*

[1] The term "LP-relative" should depend on which LP relaxation we choose but in this paper we always mean the naive LP.

Along the way, we show arc-WMFT and all three WMFT-cover problems are APX-hard even for unit capacities.

Finally we extend the known frontier of tractable WMFT instances. A *root-or-radial* instance is one in which, for some fixed root vertex, each commodity path either goes through the root, or has one of its endpoints an ancestor of the other with respect to the root. For example, every *spider* instance, where only one node has degree greater than 2, is root-or-radial.

Theorem 5. *Root-or-radial edge-WMFT instances can be solved in strongly polynomial time.*

Our proof of Theorem 5 is via a combinatorial reduction to *bidirected flow* [11]; the reduction also yields a polyhedral characterization of the feasible solutions for root-or-radial instances. It is natural to ask if the other variants we study behave similarly for root-or-radial instances and indeed, Theorem 5 holds for all 6 combinations of {arc, vertex, edge} capacities and {WMFT, WMFT-cover}. The case of arc capacities is simple since the naïve LP is totally unimodular. The specific result that arc-WMFT $\in$ P for spiders was noted already by Erlebach & Vukadinović [14].

Related Work. Edge-WMFT appears in the literature under a variety of names including *cross-free-cut matching* [18] in the unit-capacity case and *packing of a laminar family* [9]. One generalization is the *demand* version [8] in which each commodity i is given a requirement r_i and we require $y_i \in \{0, r_i\}$ for each feasible solution.

Arc- and edge-WMFT with unit capacities are equivalent to the weighted *edge-disjoint paths* (EDP) problem on trees. The hardness results we mentioned of [1, 20] are in fact hardness results for EDP. For fixed k, edge-EDP with at most k commodities is polynomial-time solvable by results of the graph minors project. See e.g. [32, §70.5] for further discussion.

The extreme points of the natural LP for edge-WMFT arise frequently in the literature of LP-based network design [8, 9, 13, 16, 17, 19, 22, 25, 34]. From this perspective, edge-WMFT is a natural starting point for an investigation of how large capacities/requirements affect the difficulty of weighted network design problems.

1.1 Formulation

For edge-WMFT, 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 $d \in D$; this is without loss of generality since the supply graph and demand edges are undirected. (In the arc-capacitated case, D is a set of arcs.) Since we discuss WMF only on 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* for edge-WMFT if it satisfies $y \geq \mathbf{0}$ and it meets the *capacity constraints*

$$\forall e \in E : \sum_{d: e \in p_d} y_d \leq c_e. \tag{1}$$

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

Later we will use the natural analogues of the above integer program for vertex- or arc-capacities, and for WMFT-cover.

1.2 Overview of Paper

In Section 2 we give the general framework, which is needed to understand most of the rest of the paper, and prove Theorem 1. In Section 3 we generalize to arc- or vertex-capacities, proving Theorem 2. In Section 4 we generalize to covering problems, proving Theorem 3. In Section 5 we give the proof of Theorem 4, the matching hardness result. In Section 6 we prove Theorem 5, showing that root-or-radial instances are in P. Concluding remarks and open problems appear in Section 7.

2 Framework and Edge-WMFT Approximation

In this section we obtain a $\min\{3, 1+4/\mu+6/(\mu^2-\mu)\}$-approximation algorithm for edge-WMFT, assuming $c_e \geq \mu \geq 2$ for each edge e. The algorithm uses the iterated rounding paradigm [22] and its extension to iterated relaxation [25]. We evolve the natural LP over iterations and ultimately develop an integral solution which has weight at least as large as that of an optimal feasible fractional solution, but is infeasible due to violating capacity constraints, albeit in a limited way. In Sections 2.1 and 2.2 we show how to compute high-weight feasible solutions from this +2-violating solution.

The natural LP relaxation of edge-WMFT, which we denote by $(\mathcal{F})$, is as follows:

$$\text{maximize } w \cdot y \text{ over } y \in \mathbb{R}^D_{\geq 0} \text{ subject to the capacity constraints (1).} \tag{$\mathcal{F}$}$$

Any integral vector y is feasible for $(\mathcal{F})$ iff it is a feasible multicommodity flow. We will use OPT to denote the optimal value of the linear program $(\mathcal{F})$. This program has a linear number of variables and constraints, and thus can be solved in polynomial time. We now give a beginning ingredient of iterated rounding/relaxation.

Lemma 6. *Let y^* be an optimal solution to $(\mathcal{F})$, define $\mathrm{OPT} = w \cdot y^*$, 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 $(\mathcal{F})$. Then $\mathrm{OPT}' = \mathrm{OPT} - tw_d$.*

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 $\mathrm{OPT}' \geq w \cdot (y^* - z) = \mathrm{OPT} - tw_d$. 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 $\mathrm{OPT} \geq \mathrm{OPT}' + tw_d$. Combining these inequalities, we are done. $\square$

We now explain our approach in general terms. It is helpful if we can assume $y_d \leq 1$ holds for each demand edge d, since this will give a good bound on the number of iterations. Indeed, this is without loss of generality using the proof of Lemma 6[2]. Hence we use $(\mathcal{F}_1)$ from now on in place of $(\mathcal{F})$:

$$\text{maximize } w \cdot y \text{ over } y \in [0,1]^D \text{ subject to the capacity constraints (1).} \tag{$\mathcal{F}_1$}$$

We iteratively build an integral solution to $(\mathcal{F}_1)$ with value at least equal to OPT. The first step in each iteration is to solve $(\mathcal{F}_1)$, obtaining solution y^*. If $y^*_d = 0$ for some demand edge d, then we can discard d without affecting the optimal value of $(\mathcal{F}_1)$. If $y^*_d = 1$ for some d, then we can route one unit of flow along p_d and update capacities accordingly. By Lemma 6, 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 7. *Suppose that y^* is an extreme point solution to $(\mathcal{F})$, 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 (1) 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, denoted IteratedFlowSolver, is given below.

Procedure IteratedFlowSolver

1: Set $\widehat{y} = \mathbf{0}$
2: If $D = \varnothing$ terminate and return $\widehat{y}$
3: Let y^* be an optimal extreme point solution to $(\mathcal{F}_1)$
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 7 and contract e
7: Go to step 2

[2] In detail, start by computing an optimal y, then route $\lfloor y_d \rfloor$ units of each commodity d and reduce the capacities accordingly. The LP drop equals the profit of the routed flow, and the new LP has an optimum with $y < \mathbf{1}$.

Assuming Lemma 7, 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 8. IteratedFlowSolver *runs in polynomial time.*

Proof. In each iteration we decrease $|D| + |E|$, so polynomially many iterations occur. Since ($\mathcal{F}$) and ($\mathcal{F}_1$) can be solved in polynomial time, the result follows. $\square$

Property 9. *The integral solution computed by* IteratedFlowSolver *violates each capacity constraint* (1) *by at most* $+2$*.*

Proof. Consider what happens to any given edge e during the execution of the algorithm. In each iteration (and in the preprocessing of ($\mathcal{F}$) to ($\mathcal{F}_1$)) 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 (1) 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 (1) for e by at most $+2$. $\square$

Property 10. *The integral solution computed by* IteratedFlowSolver *has objective value at least equal to* OPT*.*

Proof. When we contract an edge e we just remove a constraint from ($\mathcal{F}_1$), which cannot decrease the optimal value of ($\mathcal{F}_1$) since it is a maximization LP. In every other iteration and in preprocessing, Lemma 6 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 ($\mathcal{F}_1$) is 0. Thus the overall weight of flow routed must be at least as large as the initial value of OPT. $\square$

2.1 Minimum Capacity $\mu = 2$

As per Property 9, our iterated solver may exceed some of the edge capacities. When $c_e \geq 2$ for each edge e we invoke the following theorem to produce a high-weight feasible solution.

Theorem 11. *Suppose that $\widehat{y}$ is a nonnegative integral vector so that for each edge e, the constraint* (1) *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 $w \cdot y' \geq (w \cdot \widehat{y})/3$, with $\mathbf{0} \leq y' \leq \widehat{y}$, and such that y' satisfies all constraints* (1)*.*

Proof. We use the following theorem from [9, Thm. 6], which we restate in our notation:

Theorem 12 (Cheriyan, Jordán, Ravi)**.** *Consider a tree $T = (V, E)$ with nonnegative integral capacities $\{c_e\}_{e \in E}$. Let I be a collection of paths in the tree so that for each edge $e \in E$, at most $2c_e$ paths from I use e. Let $\{w_i\}_{i \in I}$ be nonnegative weights. Then there is an algorithm CJR with running time $O(|V||I|)$ to find a set $I' \subseteq I$ with $w(I') \geq w(I)/3$, and such that for each edge $e \in E$, at most c_e paths from I' use e.*

Consider the special case of Theorem 11 when $\widehat{y}$ is 0-1. Let $I = \{d \mid \widehat{y}_d = 1\}$, run CJR, and let y' be the characteristic vector of I'. This proves Theorem 11 in this special case.

In general, break $\widehat{y}$ into two parts, an *even* part $\widehat{y}^0 = 2\lfloor \widehat{y}/2 \rfloor$ and an *odd* part $\widehat{y}^1 = \widehat{y} - \widehat{y}^0$; and break the capacities into two parts, $c^0_e = \sum_{d: e \in p_d} \widehat{y}^0_d$, $c^1_e = c_e - c^0_e$. For input $(\widehat{y}^1, c^1)$ to Theorem 11 we can efficiently find the desired y'^1 using the previous paragraph; for input $(\widehat{y}^0, c^0)$ to Theorem 11 we can simply take $y'^0 = \widehat{y}^0/2$; and one sees that $y' = y'^0 + y'^1$ is a valid output for the original input $(\widehat{y}, c)$. $\square$

We now prove part (a) of Theorem 1.

Proof of Theorem 1(a). Let $\widehat{y}$ be the output of IteratedFlowSolver. Since $c_e \geq 2$ for each edge e, and since by Property 9 each edge's capacity is additively violated by at most $+2$, Theorem 11 applies. Thus y' is a feasible solution to the edge-WMFT instance with objective value $w \cdot y' \geq w \cdot \widehat{y}/3 \geq \mathrm{OPT}/3$, using Property 10. Finally, since $(\mathcal{F})$ is an LP-relaxation of the edge-WMFT problem, OPT is at least equal to the optimal edge-WMFT value, and so y' is a 3-approximate feasible integral solution. ☐

2.2 Arbitrary Minimum Capacity

Given the infeasible solution $\widehat{y}$ produced by IteratedFlowSolver, we want to reduce $\widehat{y}$ so as to attain feasibility, while losing as little weight as possible. 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. By Property 9, $f \leq \mathbf{2}$. Note now that a reduction z with $\mathbf{0} \leq z \leq \widehat{y}$ makes $\widehat{y} - z$ a feasible (integral) edge-WMFT solution if and only if z is a feasible (integral) solution to the following linear program.

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

Notice that $(\mathcal{F}_c)$ is a covering analogue of $(\mathcal{F})$ with upper bounds. Furthermore, Jain's iterated rounding framework [22] gives an LP-relative 2-approximation algorithm to find an optimal integral solution.

Theorem 13 (Jain [22])**.** *There is a polynomial-time algorithm which, when $(\mathcal{F}_c)$ is feasible, returns an integral feasible solution $\widehat{z}$ such that $w \cdot \widehat{z} \leq 2 \cdot \mathrm{OPT}(\mathcal{F}_c)$.*

Proof sketch. Jain's algorithm takes a set family and uncrosses it to get a laminar family, which has a tree structure. Here, we have the tree structure directly. We sketch some of the remaining details. In each iteration, we obtain an extreme point optimal solution z^* to the linear program $(\mathcal{F}_c)$. 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 $\mathbf{0} < z^* < \mathbf{1}$, a lemma of Jain (see also a shorter proof in [26]) 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. ☐

Here is how we use Theorem 13 to approximate edge-WMFT instances on trees.

Proof of Theorem 1(b). Notice that $z = \frac{2}{\mu+2}\widehat{y}$ is a feasible fractional solution to $(\mathcal{F}_c)$ by the definition of f. Hence, the optimal value of $(\mathcal{F}_c)$ is at most $\frac{2}{\mu+2}\widehat{y} \cdot w$. Thus the solution $\widehat{z}$ produced by Theorem 13 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 edge-WMFT problem, with $w \cdot (\widehat{y} - \widehat{z}) \geq (1 - \frac{4}{\mu+2})\widehat{y} \cdot w \geq (1 - \frac{4}{\mu+2})\mathrm{OPT}$. This gives us a $1/(1 - \frac{4}{\mu+2}) = 1 + 4/\mu + O(1/\mu^2)$ approximation algorithm for edge-WMFT.

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 $\widehat{y}' \cdot w \geq (1 - \frac{2}{\mu+2})\mathrm{OPT}$. The second round analogously extracts from $\widehat{y}'$ a solution with $+0$ violation, i.e. 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 7

First, we need the following simple inequality.

Lemma 14. *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 7. Since y^* is a basic solution with $0 < y < 1$, it follows that there exists a set $E^* \subset E$ of edges with $|E^*| = |D|$ such that y^* is the unique solution to

$$\sum_{d \in D: e \in p_d} y_d = c_e \qquad \forall e \in E^*. \tag{2}$$

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 (2), $\mathbf{0} < y^* < \mathbf{1}$, 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 14 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. $\square$

We remark that Lemma 7 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. [9], is given in Figure 1.

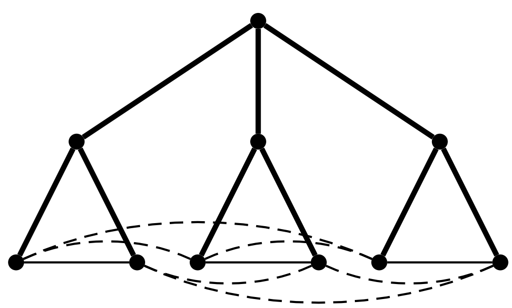

Figure 1: An extreme point solution to ($\mathcal{F}$). There are 9 edges in the supply graph, shown as thick lines; each has capacity 1. There are 9 demand edges, shown as thin lines; the solid ones have value 1/2, and the dashed ones have value 1/4. This is a tight example for Lemma 7 because each edge lies on at least three demand paths.

2.4 Extreme Point Structure for Covering Problems

Later on, we will need to deal more explicitly with the covering LP relaxation ($\mathcal{F}_c$). Hence at this point, we show that the counting lemma for the packing problem ($\mathcal{F}$) implies the same type of counting lemma for the covering analogue.

Corollary 15 (Counting Corollary for Covers)**.** *Suppose that $\mathbf{0} < z^* < \mathbf{1}$ is a extreme point solution to ($\mathcal{F}_c$). Then there is an edge $e \in E$ so that $|\{d \in D : e \in p_d\}| \leq 3$.*

Proof. Being an extreme point is the same as saying that z^* is uniquely defined by its support together with a linearly independent set of tight edges whose capacity-covering constraints are satisfied exactly. Now consider z^* as a solution to the *packing* problem with the same capacities for tight edges, and non-tight capacities set to $+\infty$. The same set of edges still have their capacity-packing constraints satisfied exactly, these constraints remain linearly independent, and z^* is clearly feasible (since for each constraint, it is either tight, or it is a packing constraint with capacity $+\infty$). Thus z^* is an extreme point solution to ($\mathcal{F}$) for the new WMFT instance. Using Lemma 7 on this instance, we are done. $\square$

3 Vertex-WMFT and Arc-WMFT

The natural LP relaxation for edge-WMFT, given in Section 1.1, admits straightforward analogues for vertex- and arc-WMFT: we replace constraint (1) with a vertex- or arc-capacity constraint. Let us denote these LPs by *vertex-*($\mathcal{F}$) and *arc-*($\mathcal{F}$). Analogously to the methods of Section 2, the crux of our work can be performed under the assumption that $y_d \leq 1$ for each d, so we similarly define *vertex-*($\mathcal{F}_1$) and *arc-*($\mathcal{F}_1$).

The key in our approaches to arc-WMFT and vertex-WMFT are analogues of Lemma 7. We defer their proofs to Section 3.1 and Section 3.2.

Lemma 16 (Vertex-WMFT Counting Lemma)**.** *Suppose that y^* is an extreme point solution to vertex-($\mathcal{F}$), and that $0 < y^*_d < 1$ for each demand edge $d \in D$. Then there is a vertex v so that $|\{d \in D : v \in p_d\}| \leq 7$.*

Lemma 17 (Arc-WMFT Counting Lemma)**.** *Suppose that y^* is an extreme point solution to arc-($\mathcal{F}$), and that $0 < y^*_d < 1$ for each demand edge $d \in D$. Then there is an arc a so that $|\{d \in D : a \in p_d\}| \leq 7$.*

As with Corollary 15, these imply counting lemmas for the analogous covering LPs. Analogous to the proof of Property 9, iterated relaxation yields:

Corollary 18. *There is a polynomial-time algorithm which, when given a vertex-WMFT or arc-WMFT instance, produces y such that $w \cdot y$ is at least as large as the optimum for the instance, and such that the y is feasible when each capacity is increased by* 6*.*

In order to proceed with our framework, we need an "LP-relative" $O(1)$-approximation for arc-WMFT-cover and vertex-WMFT-cover (in place of Jain's 2-approximation algorithm).

Corollary 19 (LP-Relative WMFT-Cover Approximations)**.** *There is a polynomial-time algorithm which, when* vertex-($\mathcal{F}_c$) *is feasible, returns an integral feasible solution z such that $w \cdot z \leq 7 \cdot \mathrm{OPT}(\text{vertex-}(\mathcal{F}_c))$, and similarly for the arc-version.*

Proof. The proof is analogous to the proof of Theorem 13, using Jain's iterated rounding framework. It suffices (say, for vertex-WMFT-cover) to show that any nonzero extreme point solution z^* to vertex-($\mathcal{F}_c$) has $z^*_d \geq 1/7$ for some d. If some $z^*_d \geq 1$ we are done. Otherwise by the vertex-analogue of Corollary 15, some tight vertex v is on at most 7 demand paths; by tightness $\sum_{d:v \in p_d} z^*_d = f_v \geq 1$. Thus some d with $v \in p_d$ has $z^*_d \geq 1/7$, as needed. $\square$

We are almost at the main result of $1 + O(1/\mu)$-approximation algorithms for arc-WMFT and vertex-WMFT. In particular for small μ, we will need a constant-factor approximation algorithm for vertex-WMFT. In fact, an LP-relative 5-approximation can be obtained by adapting the 4-approximation algorithm for edge-WMFT of Chekuri et al. [8]; we give details in the appendix. (Recall [8] also gives a 4-approximation for arc-WMFT.) With this, we have the main result of this section; the proof is analogous to Theorem 1(b) but we give a brief review for clarity.

Theorem 2. *There are $1 + O(1/\mu)$-approximation algorithms for arc-WMFT and vertex-WMFT, for all $\mu \geq 1$.*

Proof. We prove the vertex-WMFT version; the arc version is analogous. We do not attempt to optimize the constants. As in the proof of Theorem 1, it is no loss of generality to assume the additional constraints $y_d \leq 1$, hence we work with vertex-($\mathcal{F}_1$) instead of vertex-($\mathcal{F}$).

Corollary 18 gives us a solution y to vertex-($\mathcal{F}_1$) with $w \cdot y \geq \mathrm{OPT}$ and such that y violates each vertex capacity by at most +6. Let f_v equal the amount by which the capacity for v is violated, or 0 if the capacity is not violated. Apply Corollary 19 to vertex-($\mathcal{F}_c$) for this choice of f and $\widehat{y} = y$; we get a z such that $w \cdot z \leq 7 \cdot \mathrm{OPT}(\text{vertex-}(\mathcal{F}_c))$. Moreover, $y - z$ is feasible for vertex-($\mathcal{F}_1$).

Just as before, it is easy to verify that $\frac{6}{\mu+6} y$ is a feasible solution to vertex-($\mathcal{F}_c$). Hence we have

$$w \cdot (y - z) \geq w \cdot y - 7 \cdot \mathrm{OPT}(\text{vertex-}(\mathcal{F}_c)) \geq w \cdot y - 7\frac{6}{\mu+6} w \cdot y \geq \left(1 - \frac{42}{\mu+6}\right) \mathrm{OPT}(\text{vertex-}(\mathcal{F}_1)).$$

For large μ, this implies that y is an $1+O(1/\mu)$-approximately optimal solution to the vertex-WMFT instance. For small μ, we use the 5-approximation mentioned above. Combining these facts, we are done. □

3.1 Counting Lemma for Vertex Capacities

Proof of Lemma 16. Call a vertex *tight* if $\sum_{d \in D: v \in p_d} y^*_d = c_v$. Using that y^* is basic and $0 < y < 1$, it follows that there exists a set $V^* \subset V$ of tight vertices with $|V^*| = |D|$ such that y^* is the unique solution to

$$\sum_{d \in D: v \in p_d} y_d = c_v \qquad \forall v \in V^*. \tag{3}$$

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

We now introduce several properties that hold without loss of generality (that is to say, without affecting the fact that y^* is the unique solution to Equation (3)). First, for each demand edge $d = yz$, we may assume both y and z are tight, since otherwise we can replace d by $y'z'$ where y' is the closest tight vertex to y on p_d and z' is defined similarly (note, possibly $y' = z'$). Second, every degree-1 vertex is tight, since we can iteratively delete degree-1 vertices that are non-tight. Now, if v is a degree-2 vertex with neighbours u and w, define *contracting the vertex* v to mean removing v from the graph and making u, w adjacent. Third, every degree-2 vertex is tight, since we can iteratively contract degree-2 vertices that are non-tight.

Now we will apply a counting argument. Let t_i denote the number of tight vertices of degree i, and u_i denote the number of non-tight vertices of degree i. So $u_1 = u_2 = 0$ and all other values are non-negative. We give the high-level argument and then fill in the details. Say an edge uv of the supply graph is *special* if both u and v are degree-2 (tight) vertices, and let s be the total number of special edges. We re-use the notion from the proof of Lemma 7 that each demand edge $d = yz$ has two endpoints, one *owned* by y and the other *owned* by z. (I.e. the number of endpoints owned by v is equal to its degree in (V, D) where a loop counts twice to the degree.) Let $t_{\geq 3} = \sum_{i \geq 3} t_i$ and define $u_{\geq 3}$ similarly.

Claim 20. *The degree-2 tight vertices own at least $2s$ endpoints.*

Claim 21. $s \geq t_2 - (u_{\geq 3} + t_1 + t_{\geq 3} - 1)$.

Claim 22. $t_1 > t_{\geq 3} + u_{\geq 3}$.

Given these claims, we combine them as follows. Count the number w of endpoints owned by degree-1 (tight) vertices; this value satisfies

$$\begin{aligned}
w &\leq 2|V^*| - 2s && \text{by Claim 20 and since } |D| = |V^*| \\
&= 2(t_1 + t_2 + t_{\geq 3}) - 2s \\
&\leq 2(t_1 + t_2 + t_{\geq 3}) - 2(t_2 - (u_{\geq 3} + t_1 + t_{\geq 3} - 1)) && \text{by Claim 21} \\
&= -2 + 4t_1 + 4t_{\geq 3} + 2u_{\geq 3} \\
&< 8t_1 - 2. && \text{by Claim 22}
\end{aligned}$$

So $w < 8t_1$, in particular there is some degree-1 tight vertex v which owns at most 7 endpoints. Clearly $|\{d \in D : v \in p_d\}| \leq 7$, which completes the proof of Lemma 16. We now move on to the supporting claims.

Proof of Claim 20. By definition of "special edge," note that the special edges can be partitioned into inclusion-maximal paths. Let $v_0, v_1, \ldots, v_k$ be the vertices of one such path, i.e. suppose the edges $v_{i-1}v_i$ are special for $1 \le i \le k$, so each v_i is a degree-2 tight vertex. We will show that the vertices of the path own at least $2k$ endpoints; then by adding over all paths it follows that the set of all degree-2 vertices own at least $2s$ endpoints.

Define v_{-1} to be the neighbour of v_0 which is not equal to v_1 and define v_{k+1} similarly. For $1 \le i \le k$, say that a demand edge $d = yz$ has a "left endpoint at v_i" if one of y, z is equal to v_i and $v_{i-1} \notin p_d$; define right endpoints similarly. Notice that the number of endpoints owned by $\{v_i \mid 0 \le i \le k\}$ equals the total number of left and right endpoints therein.

For each $1 \le i \le k$, since the constraints (3) for tight vertices v_{i-1} and v_i are linearly independent, there is at least one demand path p_d containing exactly one of v_{i-1} and v_i. Since the vertex capacities are integral and both are tight, in fact there are at least 2 demand paths containing exactly one of v_{i-1} or v_i. Thus the number of right endpoints at v_{i-1} plus the number of left endpoints at v_i is at least 2. Adding over all k, we are done. ☐

Proof of Claim 21. Define T' to be the tree obtained from T by contracting all degree-2 tight vertices; viewing this process in reverse, T can be obtained from T' by subdividing its edges. For each edge of T', if it is subdivided $x \ge 1$ times, that corresponds to a path of $x-1$ special edges in T. Note T' has $u_{\ge 3} + t_1 + t_{\ge 3}$ vertices and thus $u_{\ge 3} + t_1 + t_{\ge 3} - 1$ edges. Hence the number of special edges is at least

$$s \ge t_2 - (u_{\ge 3} + t_1 + t_{\ge 3} - 1). \quad \square$$

Proof of Claim 22. Using the handshake lemma (as in the proof of Lemma 14) we know that $t_1 = 2 + \sum_{i \ge 3}(i-2)(t_i + u_i)$, and the desired result follows. ☐

(End of proof of Lemma 16.) ☐

3.2 Counting Lemma for Arc Capacities

Proof of Lemma 17. Let $|A^*|$ denote a maximum size linearly independent set of tight arcs, so $|D^*| = |A^*|$. Whenever both uv and vu are non-tight, we contract the edge $\{u, v\}$. Furthermore, for each edge $\{u, v\}$ such that both uv and vu are tight, subdivide uv with a new vertex w so that $c_{uw} = c_{uv}$, $c_{vw} = c_{vu}$, and consider uw and vw as tight instead of uv and vu. What results is a directed tree where every edge is tight in exactly one direction. Its vertex set V^* satisfies $|V^*| = |A^*| + 1$.

In (V^*, A^*) let n_1 be the number of degree-1 vertices, n_2 the number of degree-2 vertices, and $n_{\ge 3}$ the number of vertices of degree at least 3. Define an edge to be *special* if both endpoints have degree 2, as in the proof of Lemma 16. Analogously to Claim 21, the number of special edges satisfies $s \ge n_2 - n_1 - n_{\ge 3} + 1$. Analogously to Claim 20, we have $n_1 > n_{\ge 3}$. We will show moreover that the degree-2 vertices own at most $2s$ endpoints; then we will be done since the number of endpoints owned by degree-1 vertices is at most

$$2|A^*| - 2s = 2(n_1 + n_2 + n_{\ge 3} - 1) - 2s \le 4n_1 + 4n_{\ge 3} - 4 < 8n_1.$$

To show that the degree-2 vertices own at least $2s$ endpoints, we proceed along the lines of the proof of Claim 20. Take a sequence of vertices $v_0, v_1, \ldots, v_{k+1}$ in (V^*, A^*) such that $\delta(v_i) = 2$ iff $1 \le i \le k$. Call the set of arcs between them P; it is a path when ignoring directions, but P is not in general a dipath since the orientations of arcs on P need not be consistent. The set P contains $k-1$ special edges, and $k+1$ edges in total. We will show that the vertices $\{v_1, \ldots, v_k\}$ own at least $2k-2$ endpoints; then by adding over all such P we will have shown that the degree-2 vertices own at least $2s$ endpoints.

For convenience we call arcs in P of the form $v_i v_{i-1}$ *leftwards*, and other arcs *rightwards*. Now for any demand arc $d \in D^*$ such that p_d intersects P, either p_d intersects only leftwards arcs or rightwards arcs; thus call p_d a *leftwards* or *rightwards* demand path correspondingly. Let there be r rightwards arcs and $k+1-r$ leftwards arcs in P. We will show, using linear independence arguments similar to ones made previously, that $\{v_1, \ldots, v_k\}$ own at least $2r-2$ endpoints of rightward demand arcs and at least $2k-2r$ endpoints of leftward demand arcs, giving the desired result.

Let $v_i v_{i+1}$ and $v_j v_{j+1}$ with $k \geq j > i \geq 0$ be rightwards arcs of P such that all intermediate arcs of P are leftwards (i.e., "consecutive" rightwards arcs). By linear independence there is at least one rightwards demand path using exactly one of $v_i v_{i+1}$ or $v_j v_{j+1}$; in fact since both arcs are tight with integral capacities and y^* is fractional, there are at least two such rightwards demand paths. This implies that the vertices $\{v_u\}_{u=i+1}^{j}$ own at least 2 endpoints of rightwards demand paths. By considering all possible choices of i, j it follows that $\{v_1, \ldots, v_k\}$ own at least $2r - 2$ endpoints of rightwards demand paths; an analogous argument works for leftwards demand paths. □

4 Approximating WMFT-Cover

Recall that edge-WMFT-cover is the problem of minimizing $w \cdot z$ over $z \in \mathbb{R}^D_{\geq 0}$ so that $\sum_{d:e \in p_d} z_d \geq f_e$ for all e. Let μ denote the minimum covering requirement $\min_e f_e$. Our framework also shows that edge-WMFT-cover can be approximated within ratio $1 + O(1/\mu)$ (and the same idea works for arc- and vertex-versions, with worse constants). **Note:** WMFT-cover has no upper bounds on variables (unlike ($\mathcal{F}_c$) for example) and in fact it is not hard to see (using hardness results from the next section) such upper bounds would preclude $(1 + O(1/\mu))$-approximability.

Theorem 3. *There is a $(1 + 2/\mu)$-approximation algorithm for edge-WMFT-cover.*

Proof. First, we need an iterated relaxation algorithm for edge-WMFT-cover. Let the naive relaxation of edge-WMFT-cover be denoted by ($\mathcal{F}$)-cover.

Corollary 23. *There is a polynomial-time algorithm which produces $z \geq \mathbf{0}$ such that $w \cdot z \leq \mathrm{OPT}(\mathcal{F})$-cover and $\forall e \in E : \sum_{d:e \in p_d} z_d \geq f_e - 2$.*

Proof. We use an analogue of IteratedFlowSolver together with the counting corollary for covers, Corollary 15. In each iteration, if some $y_d = 0$ or some $y_d \geq 1$, we reduce the problem. Otherwise some tight edge lies on at most 3 demand paths. Since $y_d < 1$ for each d, $f_e \leq 2$. We reset $f_e = 0$ (i.e., discard the constraint) and continue iterating. The usual analysis completes the proof. □

To obtain Theorem 3, define $f'_e := f_e + 2$ for each edge e and run the just-mentioned algorithm on f'. The output z is a feasible solution to the original instance. Moreover, if z^* is an optimal solution to the original ($\mathcal{F}$)-cover, then $(1 + 2/\mu)z^*$ is a feasible fractional solution to the new LP. Hence $w \cdot z$, which is at most the optimum of the new ($\mathcal{F}$)-cover, is at most $(1 + 2/\mu)$ times the optimum of the original ($\mathcal{F}$)-cover. □

5 Hardness of Approximation

The goal of this section is to establish that no approximation ratio asymptotically better than $1 + O(1/\mu)$ is possible for the arc, vertex, or edge-versions of WMFT (resp. WMFT-cover) even if all profits (resp. costs) are unit; we use MFT in place of WMFT to indicate that all weights/profits are unit. All of our reductions are modeled closely off of a construction of Garg et al. [18, Thm. 4.2], which they used to show that edge-MFT is APX-hard. Since we use and adapt it so much, it is useful to review it here, in a slightly simpler form.

Theorem 24 (Garg, Vazirani & Yannakakis [18])**.** *Edge-MFT is* APX*-hard.*

Proof. The reduction is from *3-bounded maximum three-dimensional matching* (MAX 3DM–3); an instance consists of three disjoint sets X, Y, Z and a family $S \subset X \times Y \times Z$ of triples such that each element is in at most 3 triples. The objective is to find a maximum-size disjoint set of triples from S. We let $n = |S|$ and it is not hard to see $|X|, |Y|, |Z| \leq n$ without loss generality.

Kann [23] 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 $n/7$. Hence it is NP-hard to additively approximate MAX 3DM–3 within $n\delta/7$.

Here is the reduction. On input X, Y, Z, S our tree has a root r, a first level of nodes adjacent to r in bijection with $X \cup Y \cup Z$, and a second level of size $2n$. We abuse notation and let $x \in X$ stand both for an element of X and the corresponding node in the first level of the tree, and similarly for Y, Z. Then, denoting[3] the triples S as $\{s_i = (x_i, y_i, z_i)\}_{i=1}^n$, for each i the tree has two nodes p_i, q_i adjacent to node y_i. The edges between r and Y have capacity 2 and all other edges have capacity 1. Finally, the set of demands is $D = \{(x_i, p_i), (p_i, q_i), (q_i, z_i) \mid 1 \leq i \leq n\}$. This completes the edge-MFT instance description; we illustrate in Figure 2.

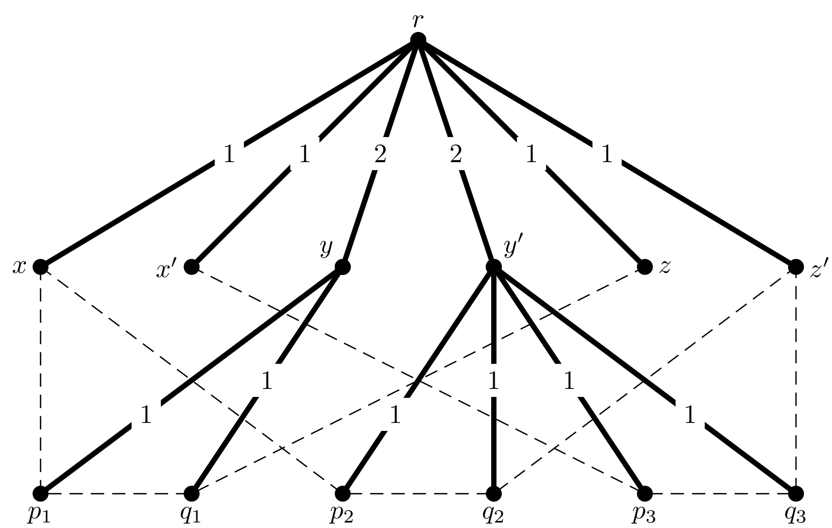

Figure 2: Illustration of the reduction; here $X = \{x, x'\}$ (similarly for Y, Z) and the triples are $(x, y, z), (x, y', z'), (x', y', z')$. The tree is denoted by thick edges with capacities shown on each edge, whereas the demands are dashed thin lines.

Claim 25. *We can convert a set of t disjoint triples in S to a feasible multicommodity flow of value $n + t$ in polynomial time, and vice-versa.*

Thus the optimal value of the MFT instance is n more than the optimal value of the 3DM instance.

Proof. First, consider a set of disjoint triples, denoted $\{s_i \mid i \in I\}$ for some $I \subset \{1, \ldots, n\}$. For $i \in I$ assign unit flow to demands (x_i, p_i) and (q_i, z_i); for $i \notin I$ assign unit flow to demand (p_i, q_i); assign zero to all other flows. Then it is easy to verify the resulting multicommodity is feasible, and since it assigns a total amount $2|I| + (n - |I|) = n + |I|$ of flow, giving the forwards direction of the proof.

Now we consider the reverse direction, converting an optimal multicommodity flow to a disjoint collection of triples. For each i, since the edges incident to p_i and q_i have unit capacity, the demands given a unit of flow are either a subset of $\{(x_i, p_i), (q_i, z_i)\}$, or (p_i, q_i). In case at most one of these three commodities is routed, we can change it to just (p_i, q_i) without violating any edge capacities. Hence WOLOG the routed demands for triple i are either (x_i, p_i) and (q_i, z_i), or just (p_i, q_i). Then the argument of the previous paragraph can be reversed, completing the proof. ∎

[3] In our notation, x_i is simply the member of X in the ith triple, so $x_i = x_j$ does not imply $i = j$.

With Claim 25 we are now basically done. We know it is NP-hard to additively approximate MAX 3DM–3 within $n\delta/7$, and consequently that the edge-MFT instance is NP-hard to additively approximate within the same threshold. Moreover, the edge-MFT optimum is at most $2n$. So it is NP-hard to multiplicatively approximate edge-MFT within ratio $1 + \delta/14$. $\square$

Now we give the trick to treat asymptotic hardness of edge-MFT depending on minimum capacity; we deal with the other problems afterwards.

Theorem 26. *For some $\epsilon > 0$, for all positive integers μ, it is* NP*-hard to approximate edge-MFT with ratio $1 + \epsilon/\mu$, even restricted to instances where all capacities are at least μ.*

Proof. Let (V, E) denote the tree in the construction of the proof of Theorem 24; it satisfies $|E| = |X| + |Y| + |Z| + 2n \leq 5n$. 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 demand edge $u'v$ to D; (3) increase the capacity of uv by μ and set the capacity of uu' to be μ.

Any solution y to the modified MFT instance can be altered so that $y_{u'v} = \mu$ for each new demand edge $u'v$, without reducing its objective value: repeatedly increase $y_{u'v}$ by 1 and reduce the value of any other flow through uv by 1. It then follows that the optimal value of the modified instance is $t+n+\mu|E|$. Furthermore, since $t+n+\mu|E| \leq (5\mu+2)n$, approximating the MFT instance to a ratio less than $1+\frac{n\delta/7}{(5\mu+2)n} = 1+\Theta(1/\mu)$ is NP-hard. $\square$

As claimed in the introduction, the same result as Theorem 26 holds for covering variants, and/or with arc- or vertex-capacities. In the rest of the section we show the ideas needed to prove all of these variants.

Vertex-MFT. Hardness follows immediately from the edge-MFT hardness, using the subdividing trick in the introduction, and giving non-subdivision nodes infinite capacity.

Whereas edge-WMFT and vertex-WMFT are polynomial-time solvable [18] if all c are equal to 1, our reductions for the other four versions will show they are APX-hard for $c = \mathbf{1}$.

Arc-MFT. To get APX-hardness we use the same construction as before, except now the *directed* demands are $\{(x_i, p_i), (q_i, p_i), (q_i, z_i)\}_i$, and all capacities are unit. As before, any flow can be converted without loss into one which either routes (q_i, p_i) or both of $(x_i, p_i), (q_i, z_i)$. The trick to get $1 + \Omega(1/\mu)$ hardness is basically the same as before except we add a new leaf u' for each *arc*, not just for each edge.

Edge-MFT-Cover. Here we reduce from *2-regular minimum three-set cover*: given a collection of triples on a ground set X, where each $x \in X$ appears in exactly 2 triples, find a min-size subcollection whose union is the whole ground set. This problem is the same as vertex cover in cubic graphs, for which a $\frac{100}{99}$ hardness ratio is known [10]. The neighbours of the root in the tree correspond the ground set; and for the ith triple $\{x_i, x'_i, x''_i\}$ we add two children p_i, q_i to x'_i, with $D = \{(x_i, p_i), (p_i, q_i), (q_i, x''_i)\}_i$. We then use the same sort of arguments as before to get APX-hardness (details also appear in [29]) and $1 + \Omega(1/\mu)$-hardness for all $\mu \geq 1$.

Vertex-MFT-Cover. To get APX-hardness with unit capacities, we use edge-MFT-cover hardness, and simply subdivide each edge by a new node. The case of higher μ is straightforward, by giving each vertex a new twin, and adding a commodity joining each vertex to its twin.

Arc-MFT-Cover. This reduction is somewhat different than the others so we express in more detail; we could not find any extremely simple proof. A graph is *properly 3-edge-coloured* if we are given a map $E \to \{\text{red, green, blue}\}$ with at most one edge of each colour incident at each vertex.

Proposition 27. *The following problem is* APX*-hard: find a minimum vertex cover in a properly 3-edge-coloured graph.*

Proof. We start from APX-hardness of vertex cover in cubic graphs. Every cubic graph (V, E) has an improper 3-edge-colouring with at most 2 edges per colour at each vertex (e.g., take a greedy proper 5-edge-colouring and coalesce colour classes). Let $U \subset V$ denote those vertices which are improper under this colouring. For each vertex $v \in U$, replace it via the transformation depicted in Figure 3. This gives a new properly 3-edge-coloured graph G'.

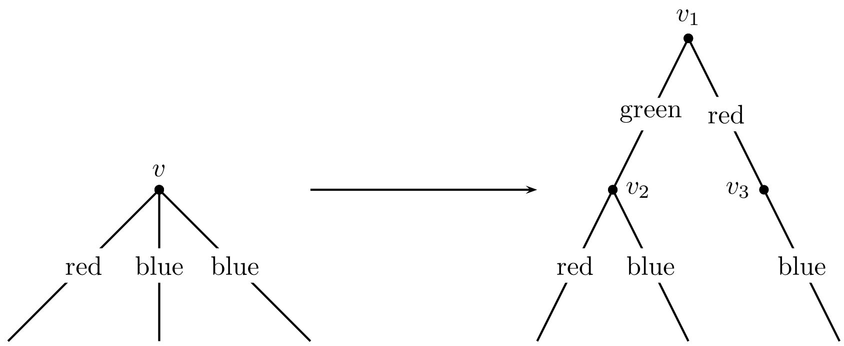

Figure 3: In (a) we see a node where the 3-edge-colouring is not proper. We replace it with the configuration in (b). It can be used for APX-hardness reduction since any vertex cover of the new graph can be modified without a size increase to either contain just v_1, or v_2 and v_3, out of the v_i's.

If τ denotes the minimum size of a vertex cover, we claim $\tau(G) + |U| = \tau(G')$. To see this, first note that for any vertex cover $X \subset V$ of G, the set $X' := X \backslash U \cup \{v_1 \mid v \notin U\} \cup \{v_2, v_3 \mid v \in U\}$ is a vertex cover of G'. Likewise, for any vertex cover X' of G', without increasing its size we may assume that for each $v \in U$, either $X' \cap \{v_1, v_2, v_3\}$ equals $\{v_1\}$ or it equals $\{v_2, v_3\}$; then the map $X \to X'$ can be reversed and it clearly yields a vertex cover of G of size $|X'| - |U|$.

Since the above reductions are polynomial-time, and since $\tau(G) \geq |V|/3 \geq |U|/3$, we find that a $(1+\epsilon)$-approximation algorithm for τ on the family of graphs G' implies a $(1 + 4\epsilon)$-approximation algorithm for τ on the family of graphs G. By APX-hardness of the latter, we are done. ☐

Now, the problem established as APX-hard in Proposition 27 is isomorphic to the following special case of set cover: *3-dimensional* (the ground set is 3-coloured as $X \uplus Y \uplus Z$ and no set contains two similarly-coloured elements) *2-regular* (every ground set element appears in exactly 2 sets) *set cover with all sets of size 2 or 3*. From here we will proceed very similarly to the arc-MFT construction.

Specifically, define the nodes $r, x_i, y_i, z_i, p_i, q_i$ as before (sets of size 2 not containing a Y element do not entail nodes p_i, q_i). For a set $\{x_i, y_i, z_i\}$ of size 3 the commodities are $\{(x_i, p_i), (q_i, p_i), (q_i, z_i)\}_i$ as before; for sets of size 2 the commodities are respectively $\{(x_i, z_i)\}$, or $\{(x_i, p_i), (q_i, p_i), (q_i, r)\}$, or $\{(r, p_i), (q_i, p_i), (q_i, z_i)\}$ when there is no element from Y, or Z, or X. Additionally, since none of these commodities can cover the arcs of the forms $(r, x_i), (y_i, q_i), (p_i, y_i), (z_i, r)$ we also introduce one commodity for each such arc. Finally, standard analysis establishes that arc-MFT-cover is APX-hard on this family of instances (in particular it's important that the last set of commodities has linear cardinality in proportion to the optimum, which is due to the $O(1)$-regularity). Finally, from here $1 + \Omega(1/\mu)$-hardness for minimum capacity μ follows from the standard argument.

6 Exact Solution for Spiders

In this section we show that edge-WMFT 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 edge-WMFT when (V, E) is a spider, each demand path p_d either goes through the root, or else lies within a single leg. We generalize this observation into the following definition.

Definition 28. *Consider an instance of edge-WMFT 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 a vertex of p_d;*

radial, *if it is not root-using and one endpoint of d is a descendant of the other with respect to root r.*

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

So for example spider instances are always root-or-radial. Nguyen [27] showed that edge-WMFT instances where all demand paths are root-using can be solved via matching.

We will give a general framework to exactly solve WMFT and its variants on root-or-radial instances. We first treat the simpler case of arc capacities. Note, Erlebach & Vukadinović [14] already proved the special case of Proposition 29 for arc-WMFT in spiders, using the same reduction; we discovered this fact only after re-discovering the reduction independently.

Proposition 29. *Arc-WMFT and arc-WMFT-cover for root-or-radial instances can be solved exactly in polynomial time.*

Proof sketch. We make two copies of the the tree, one directed out from r, and one directed in to r. For $v \in V(T)$ let $v_{\text{in}}, v_{\text{out}}$ be its respective copies, and merge r_{in} and r_{out}; note the original bidirected tree's arcs correspond to the arcs of the new directed tree T'. Now we build a circulation instance. For definiteness say we are solving an arc-WMFT instance; arc-WMFT-cover is similar. Each old arc retains its capacity as an upper bound, with no cost and lower bound 0. If $d = (u, v)$ is root-using, introduce an arc from v_{out} to u_{in}. For a radial demand edge, say from u to v with u an ancestor of v (the other case is similar), we introduce an arc from v_{out} to u_{out}. We illustrate in Figure 4.

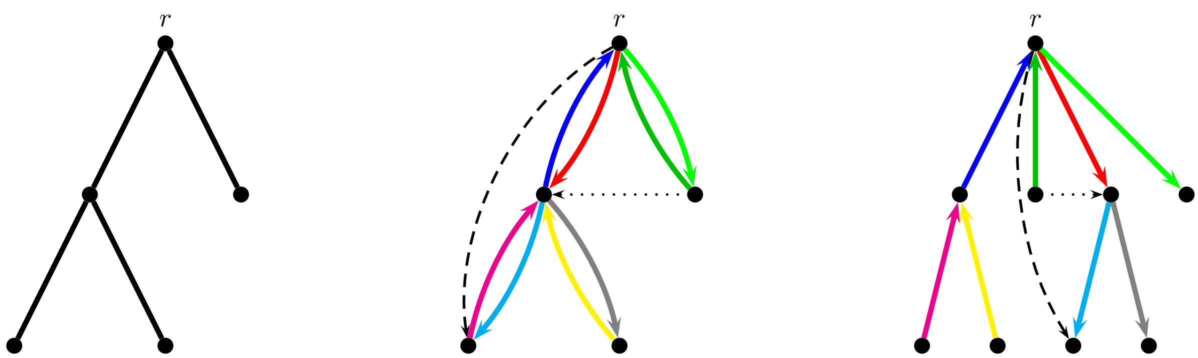

Figure 4: Left: the undirected tree for the instance. Center: the bidirected tree T_b. Right: the tree T'. The arcs of T_b and T' are coloured to show their bijection to one another. We show an out-radial demand (dashed) and a root-using demand (dotted), which correspond to dipaths in T'.

In either case the new arc gets cost w_d, no upper bound and lower bound 0. Then it is easy to see a max-cost circulation corresponds to a maximum flow in the original instance. $\square$

We observe that the same reduction makes it clear that the naïve LP relaxation is a *network matrix* and hence totally unimodular [32, §13.3].

We now move on to the more challenging case of edge-capacities; vertex-capacities will be similar but a little more complex. *Bidirected flows* were introduced by Edmonds and Johnson [11] and are a common generalization of matching and flow. Bidirected flow problems can be solved via a combinatorial reduction to b-matching (e.g., see [32]) which increases the instance size by a constant factor. We now present a reduction from root-or-radial edge-WMFT 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 max-weight bidirected flow (with upper and lower bounds) 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 5. 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 5 for an illustration.

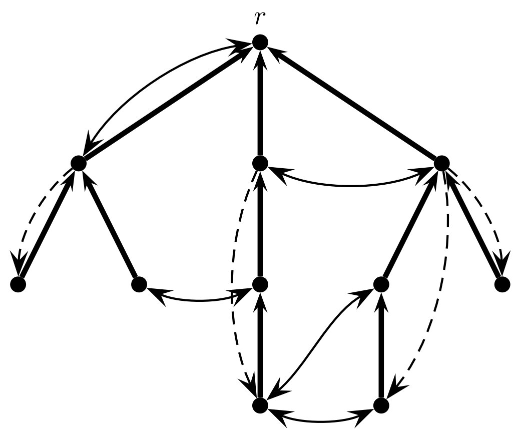

Figure 5: A root-or-radial edge-WMFT instance. The tree graph (V, E) is depicted using thick lines, and the demand edges D are thin. Radial demand edges are dashed and root-using demand edges are solid. The root is r. An arrowhead denotes a positive endpoint, while the remaining endpoints are negative; these signs correspond to the reduction in the proof of Theorem 5.

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), i.e. we conserve flow except at r. 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 $C(d)$, its characteristic vector $x = \chi^{C(d)}$ satisfies the flow conservation constraint (5). Moreover, any linear combination of these vectors is easily seen to satisfy (5), and the following converse holds.

Claim 30. *Any x satisfying (5) is a linear combination of characteristic vectors of demand cycles.*

Proof. 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 $x' = \mathbf{0}$. Then $x = x' + \sum_d x_d \chi^{C(d)} = \sum_d x_d \chi^{C(d)}$, which proves Claim 30. ∎

By Claim 30, 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 capacity constraints (1) (and nonnegativity constraints). In other words, feasible integral flows x correspond bijectively to feasible integral solutions y for the edge-WMFT 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 [2], the proof of Theorem 5 is complete. ☐

6.1 Edge-WMFT Polyhedron for Root-or-Radial Instances

The reduction used in the proof of Theorem 5 can also be used to derive the following polyhedral characterization; note that it is independent of which vertex is the root.

Theorem 31. *The convex hull of all integral feasible solutions in a root-or-radial edge-WMFT problem has the following description:*

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

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

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

Proof. 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 [32], and deals with the special case of bidirected flow when $a = b$ and $\ell = \mathbf{0}$.

Proposition 32. *Let σ denote the signs of a bidirected graph. The convex hull of the integer solutions to*

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

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

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

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

In order to apply Proposition 32 to the construction in the proof of Theorem 5, 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 root-using 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 32 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:

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

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

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

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

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

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 $r \notin U^*_F$. 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

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

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

6.2 Covering and Vertex-Capacitated Variants

By using lower-bounds instead of upper-bounds on arcs, we can modify the above construction to get a polynomial-time algorithm, and explicit LP, for edge-WMFT-cover. Moreover, vertex-WMFT and vertex-WMFT-cover also fall in this framework. To see this, move the capacity of each node in $V \backslash \{r\}$ onto its parent edge and move the capacity of r on to the loop at r. For each radial demand $d = (u, v)$ if u is an ancestor of v (resp. vice-versa), change the demand to $(\text{parent}(u), v)$ (resp. $(u, \text{parent}(v))$). The key observation now is that for each demand, the vertices that its demand path previously passed through correspond to the arcs in $C(d) \backslash \{d\}$.

6.3 Applications of Exact Formulations

Interestingly, results of Garg et al. [18, ICALP version] show that (8)–(10) is also integral in *unit-capacity* edge-WMFT. In fact they show that an intermediate LP between the naive one and (8)–(10) is integral (one requiring (10) only for subsets of edges forming a star of odd size). Tangentially, we observe their results imply a good approximation for another special case of edge-WMFT.

Proposition 33. *There is a 3/2-approximation algorithm for edge-WMFT when all edges have the same capacity.*

Proof. Let μ be the common capacity. Note that if we scale all the capacities down to unit, the optimal fractional solution goes down by a μ factor. Moreover, any integral unit flow can be scaled up by a factor μ to give a feasible flow for capacities μ. Thus, it is enough to show that for unit capacities, the optimal integral flow has at least $\frac{2}{3}$ the value of the optimal fractional flow, since unit-capacity instances can be solved in polynomial time.

To prove this, we show that any fractional flow for unit capacities, when scaled down by $\frac{2}{3}$, gives a convex combination of integral flows. In other words, we must show a solution meeting (8) and (9) can only violate (10) by a $\frac{3}{2}$ factor, when $c = \mathbf{1}$. First, (10) is vacuous for $|F| = 1$. For other odd $|F| \geq 3$, (9) implies $\sum_{d \in D} y_d \lfloor |p_d \cap F|/2 \rfloor \leq |F|/2$. Since $\frac{|F|}{2} \leq \frac{3}{2} \lfloor \frac{|F|}{2} \rfloor$ for odd $|F| \geq 3$, we are done. ☐

It is possible to synthesize our Theorems 5 and 31 with corresponding results of [18] for the unit-capacity case, by "gluing" root-or-radial instances at capacity-1 edges. Suppose for every pair of vertices with degree ≥ 3, their path contains a capacity-1 edge. Then we find again that (8)–(10) is integral. Since this result is pretty esoteric we omit the details.

7 Closing Remarks

Caprara and Fischetti [6] gave a strongly polynomial-time algorithm to separate over the family (10) of inequalities. Is the polyhedral formulation (8)–(10) useful in designing a better approximation algorithm for edge-WMFT? One roadblock is that normal uncrossing techniques seem to fail on that LP.

There is a close relation between WMFT 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 IteratedFlowSolver and Cor. 3.5 of [8], we obtain a $1 + O(\sqrt{r_{\max}/\mu})$ approximation for demand WMFT where $r_{\max}$ is the maximum demand. Shepherd and Vetta [33] showed that when the tree is a star, the $O(r_{\max})$-relaxed integrality gap of the demand analogue of $(\mathcal{F})$ is 1, and this approach extends to an approximation algorithm with ratio $1+O(r_{\max}/\mu)$ [30] for demand WMFT on stars (also known as the *demand matching problem* [33]). It would be interesting to investigate similar results on arbitrary trees.

Demand WMFT-*cover* falls in a general framework [7] wherein LP-relative approximation can be reduced to LP-relative approximation of two special cases: the non-demand version, and the *priority* version.

When every edge has capacity 1, edge-WMFT is exactly solvable [18], and Theorem 1(a) gives a 3-approximation when there are no capacity-1 edges. Can we combine these results to improve upon the 4-approximation by Chekuri et al. [8] for general instances?

A sensible generalization of the problems we have considered is {arc, vertex}-WMFT, where there are capacities on both arcs and vertices. The method of Chekuri et al. [8] gives a constant-factor approximation for this problem, however we were not able to derive a counting lemma (like Lemma 7) in this case, and consequently we are not sure if a $1 + O(1/\mu)$-approximation exists.

Acknowledgement

We would like to thank Joseph Cheriyan, Jim Geelen, András Sebő, and Chaitanya Swamy for helpful discussions.

References

[1] M. Andrews, J. Chuzhoy, V. Guruswami, S. Khanna, K. Talwar, and L. Zhang. Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs. *Electronic Colloquium on Computational Complexity*, 14(113), 2007. Preliminary versions appeared in *Proc. 46th FOCS*, pages 226–244, 2005 and *ECCC* 13(141), 2006.

[2] R. P. Anstee. A polynomial algorithm for b-matchings: An alternative approach. *Inf. Process. Lett.*, 24(3):153–157, 1987.

[3] N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree bounded directed network design. In *Proc. 40th STOC*, pages 769–778, 2008.

[4] J. Beck and T. Fiala. "Integer-making" theorems. *Discrete Applied Mathematics*, 3(1):1–8, 1981.

[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] D. Chakrabarty, E. Grant, and J. Könemann. On column-restricted and priority covering integer programs. In F. Eisenbrand and F. Shepherd, editors, *Integer Programming and Combinatorial Optimization*, volume 6080 of *Lecture Notes in Computer Science*, pages 355–368. Springer Berlin / Heidelberg, 2010.

[8] 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. Preliminary version appeared in *Proc. 30th ICALP*, pages 410–425, 2003.

[9] J. Cheriyan, T. Jordán, and R. Ravi. On 2-coverings and 2-packings of laminar families. In *Proc. 7th ESA*, pages 510–520, 1999.

[10] M. Chlebík and J. Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theor. Comput. Sci.*, 354(3):320–338, 2006. Preliminary version appeared in *Proc. 14th FCT*, pages 27–38, 2003.

[11] 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, *Combinatorial structures and their applications (Proc. 1969 Calgary Conference on Combinatorial Structures and Their Applications)*, pages 89–92. Gordon and Breach, 1970.

[12] T. Erlebach. Approximation algorithms for edge-disjoint paths and unsplittable flow. In E. Bampis, K. Jansen, and C. Kenyon, editors, *Efficient Approximation and Online Algorithms*, pages 97–134. Springer, 2006.

[13] 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.

[14] T. Erlebach and D. Vukadinović. Path problems in generalized stars, complete graphs, and brick wall graphs. *Discrete Appl. Math.*, 154:673–683, March 2006. Preliminary version appeared in Proc. 13th FCT, pages 483–494, 2001.

[15] 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.

[16] 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.

[17] H. N. Gabow, M. X. Goemans, É. Tardos, and D. P. Williamson. Approximating the smallest k-edge connected spanning subgraph by LP-rounding. *Networks*, 53(4):345–357, 2009. Preliminary version appeared in *Proc. 16th SODA*, pages 562–571, 2005.

[18] 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.

[19] M. X. Goemans. Minimum bounded degree spanning trees. In *Proc. 47th FOCS*, pages 273–282, 2006.

[20] V. Guruswami, S. Khanna, R. Rajaraman, F. B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *J. Comput. Syst. Sci.*, 67(3):473–496, 2003. Preliminary version appeared in *Proc. 31st STOC*, pages 19–28, 1999.

[21] 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.

[22] 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.

[23] V. Kann. *On the Approximability of NP-complete Optimization Problems.* PhD thesis, Royal Institute of Technology Stockholm, 1992.

[24] R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. V. Vazirani, and V. V. Vazirani. Global wire routing in two-dimensional arrays. *Algorithmica*, 2:113–129, 1987.

[25] 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.

[26] V. Nagarajan, R. Ravi, and M. Singh. Simpler analysis of lp extreme points for traveling salesman and survivable network design problems. *Operations Research Letters*, 38(3):156 – 160, 2010.

[27] T. Nguyen. On the disjoint paths problem. *Oper. Res. Lett.*, 35(1):10–16, 2007.

[28] B. Peis and A. Wiese. Throughput maximization for periodic packet routing on trees and grids. In *Proc. 8th WAOA*, 2010. To appear. Preliminary version appeared as a technical report 013-2010 at COGA, TU Berlin.

[29] D. Pritchard. k-edge-connectivity: Approximation and LP relaxation. In *Proc. 8th WAOA*, pages 225–236, 2010.

[30] D. Pritchard and D. Chakrabarty. Approximability of sparse integer programs. *Algorithmica*, 2010. In press. Preliminary versions at arXiv:0904.0859 and in Proc. 17th ESA, pages 83–94, 2009.

[31] P. Raghavan and C. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[32] A. Schrijver. *Combinatorial optimization.* Springer, New York, 2003.

[33] 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.

[34] 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 Approximation for Vertex-WMFT

Proposition 34. *There is a 5-approximation algorithm for vertex-WMFT.*

Proof. We assume the reader is familiar with the approach in [8]. Now redefine Binned Tree Colouring so that we must respect vertex capacities instead of edge capacities, and where we change the limit on the number of bins at a leaf vertex v to $n_v \leq c_v$.

Now we claim that [8, Thm. 2.2] holds with $5k$ colours instead of $4k$. The main thing to check is that we can complete the partial colouring returned by the recursive call — checking these details pertains to the last paragraph of their proof. A demand edge e joining v_i and v_j cannot use

- any colour already assigned within e's bin at v_i, of which there are less than $2k$
- any colour already assigned within e's bin at v_j, of which there are less than $2k$
- any colour which is already assigned to c_v edges passing through v, of which there are less than k, since at most $k \cdot c_v$ edges pass through v.

Hence one of the $5k$ colours is still available for colouring of e, as needed. ☐