Multicommodity Demand Flow in a Tree and Packing Integer Programs*

Chandra Chekuri[†] Marcelo Mydlarz[‡] F. Bruce Shepherd[§]

September 22, 2006

Abstract

We consider requests for capacity in a given tree network $T = (V, E)$ where each edge of the tree has some integer capacity u_e. Each request consists of an integer demand d_f and a profit w_f which is obtained if the request is satisfied. The objective is to find a set of demands that can be feasibly routed in the tree and which provide a maximum profit. This generalizes well-known problems including the knapsack and b-matching problems.

When all demands are 1, we have the integer multicommodity flow problem. Garg, Vazirani, and Yannakakis had shown that this problem is NP-hard and gave a 2-approximation algorithm for the cardinality case (all profits are 1) via a primal-dual algorithm. Our main result establishes that the natural linear programming relaxation has a constant factor gap, a factor of 4. Our proof is based on colouring paths on trees and this has other applications for wavelength assignment in optical network routing.

We then consider the problem with arbitrary demands. When the maximum demand $d_{\max}$ is at most the minimum edge capacity $u_{\min}$, we show that the integrality gap of the LP is at most 48. This result is obtained by showing that the integrality gap for the demand version of such a problem is at most 11.542 times that for the unit demand case. We use techniques of Kolliopoulos and Stein to obtain this. We also obtain, via this method, improved algorithms for the line and ring networks. Applications and connections to other combinatorial problems are discussed.

Keywords: integer multicommodity flow, tree, integrality gap, packing integer program, approximation algorithm.

1 Introduction

Let $T = (V, E, u)$ be a capacitated tree network, where each edge capacity u_e is an integer. T is termed the *supply graph* and throughout we let n denote $|V|$. We are also given a collection of demands which is encoded as a multigraph $H = (V, F, d, w)$ where each *demand* edge $f \in F$ has an associated integer value d_f and a nonnegative rational *profit* w_f. H is termed the *demand graph*.

A subset $S \subseteq F$ is *routable* (in T) if the demands can be simultaneously routed without violating any edge capacity of the tree. The DEMAND FLOW PROBLEM (DFP) is to find a routable subset S

*A preliminary version of this paper appeared in the *Proc. of ICALP*, LNCS 2179, Springer, 2003.

[†]Dept. of Computer Science, University of Illinois, Urbana, IL 61801. Email: chekuri@cs.uiuc.edu. This work was done while the author was at Lucent Bell Labs, Murray Hill, NJ.

[‡]Computer Science Dept., Rutgers University, Piscataway, NJ 08854-8019. Email: marcem@cs.rutgers.edu.

[§]Bell Labs, 600 Mountain Ave, Murray Hill, NJ 07974. Email: bshep@research.bell-labs.com.

which maximizes $w(S)$. The term "demand flow" refers to the presence of the demand values d_f which must be fully satisfied to obtain the corresponding profit w_f. Our focus is the all-or-nothing aspect to obtaining profit, rather than, for example, unsplittability of the flows. On a tree, of course, demand flows and unsplittable flows coincide since there is a unique path between every pair of nodes. In general graphs, we may or may not require extra properties on our flows such as unsplittability. As an example, we do not know the status (with respect to approximability) of the maximum demand flow problem, where demands can be satisfied by fractional flows. We discuss this in some detail at the end of the introduction.

A natural linear programming, or LP, relaxation for the demand flow problem on a tree is given below and is denoted by (LP). Here x_f denotes the percentage of demand f being satisfied, and $\text{Path}(f)$ denotes the unique path in T joining the endpoints of f.

$$\max \sum_{f \in F} w_f x_f \qquad \text{s.t} \tag{1}$$

$$\sum_{f: e \in \text{Path}(f)} d_f x_f \leq u_e \qquad \forall e \in E \tag{2}$$

$$x_f \leq 1 \qquad \forall f \in F \tag{3}$$

$$x_f \geq 0 \qquad \forall f \in F. \tag{4}$$

Obviously, the demand flow problem is modelled by adding the constraint $x_f \in \{0, 1\}$ for each demand edge f.

Our main focus is to study the integrality gap for this linear program. We establish, for the first time, that it has an $O(1)$ (approximately 48) factor gap if we assume that each demand is bounded above by the minimum capacity of an edge in the tree. A trivial lower bound of 3 on the gap is obtained by considering a star with three edges, see Figure 1.

We now review some of the well-known combinatorial optimization problems that arise as restricted versions of the demand flow problem on the tree. We first discuss the case where all demands are 1, that is, the *integer multicommodity flow problem.*

Unit Demands:

(1) *Tree is a path; Unit capacities:* Suppose that the tree is just a path and each of its edges has unit capacity. We are then essentially looking for a maximum weight stable set in an interval graph. Namely, the graph contains a node for each demand f, and two nodes are adjacent if their corresponding paths on the line intersect. It is well known that such graphs are perfect. The maximum weight stable set is then equal to a minimum cost clique cover, and both objects may be found efficiently by means of dynamic programming.

(2) *Tree is a path; Arbitrary capacities:* In this case, note that the LP formulation above is defined by a constraint matrix where each column has its ones appearing contiguously. Such matrices are totally unimodular and hence every basic solution is integral [22]. Thus, the problem can be solved in polynomial time by linear programming. It can also be solved combinatorially as a minimum cost circulation problem [8].

(3) *Tree is a star; Unit capacities:* Next suppose that the tree is a star. That is, there are nodes $v, v_1, \ldots, v_{n-1}$ and $E = \{vv_i \mid i = 1, 2, \ldots, n-1\}$. Consider the graph G obtained from H by deleting v and replacing any edge $e = vv_i \in F$ by a leaf edge $v_i v_e$ where v_e is a new node. Then a set of demand edges is routable in T if and only if they form a matching in the graph G.

(4) *Tree is a star; Arbitrary capacities:* Similar to the above arguments, a set of edges is feasible

if and only if they form a b-matching in G, where $b(v_i) = u_{vv_i}$. Thus, we may solve all problems on the star in polynomial time via matching algorithms.
(5) *Arbitrary tree:* This case is a maximum profit integer multicommodity flow problem. The special case where all profits are 1 was studied by Garg, Vazirani, and Yannakakis [15] where they gave a primal-dual algorithm which yields a factor 2 approximation. Further, they gave a polynomial time algorithm if all capacities are 1. They also showed that the problem is APX-hard even for instances with capacities 1 and 2 and trees of depth 3. For the problem with general profit weights, there seems to have been no previous constant bound known, although Cheriyan, Jordan, and Ravi [13] show that any half-integral solution to (LP) is at most a factor of $\frac{3}{2}$ times the optimal integral solution. In fact, they conjecture that the integrality gap for (LP) is $\frac{3}{2}$ for the unit demand case. We remark that the integrality gap for (LP) is lower bounded by $\frac{3}{2}$ for the unit profit case even when the tree is a star [15].

General Demands: We now consider what happens as we introduce demands to the problem, that is, when we shift from integer multicommodity flows to multicommodity demand flows.
(6) *Tree is an edge:* If the tree is itself a single edge, then the demand flow problem is precisely the knapsack problem. The relaxation (LP) is then well-known to have an integrality gap of 2. A fully polynomial time approximation scheme is also well-known for the knapsack problem. The knapsack problem is at the core of exact methods for solving integer programming since the feasible region is contained in the intersection of multiple knapsack polytopes. It is then important to establish forcing relationships on variables and cutting planes for knapsack problems as part of these methodologies. These tasks have normally been carried out on individual knapsacks separately. The demand flow problem can be seen as a collection of knapsack problems (one for each edge of the tree T) which share many variables. The study of demand flows in a tree is then a partial response to a longstanding call for more study on the interaction of several knapsacks simultaneously ([25]).
(7) *Tree is a star:* In this case, we have the demand matching problem where one is given a graph with capacities b_v for each node, and demands d_e for each edge. A set of edges M is a *demand matching* if for each node v: $\sum_{e \in M \cap \delta(v)} d_e \leq b_v$ ($\delta(v)$ denotes the edges of G incident to v). Shepherd and Vetta [24] studied this problem and show that this problem is APX-hard and that the integrality gap for (LP) is between 3 and 3.264 (2.5 and 2.764 for bipartite graphs).
(8) *Tree is a path:* In Chakrabarti et al. [9] an example is given where the supply graph is a path, but the gap between (LP) and the optimum demand flow is $\Omega(n)$. They are able to establish, however, that if we restrict to instances where $d_{\max} \leq u_{\min}$, then the integrality gap is $O(1)$. In Figure 1, it is shown that the integrality gap is at least 2.5 even with the restriction.

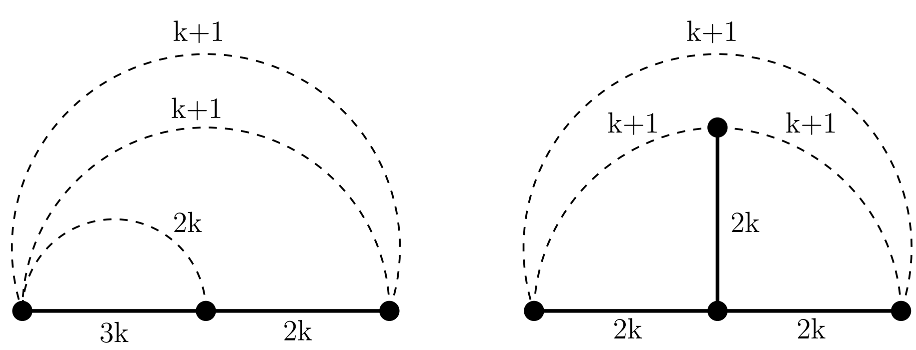

Figure 1: Integrality gap of 2.5 on a line and 3 on a star. Demand graph is shown by dashed lines; all profits are 1.

Contribution of this paper: Our main result establishes that the natural LP formulation for integer multicommodity flow has a constant factor integrality gap in the case of a tree, that is, for the demand flow problem where all demands are 1 (see the setting (5)). As mentioned above, a factor of 2 was already shown for the cardinality case in [15]. In particular, in Section 2 we prove the following:

Theorem 1.1 *Let $T = (V, E, u)$ and $H = (V, F, d, w)$ describe an instance of the maximum profit integer multicommodity flow problem on a tree. Then if (LP) has a feasible solution x of value O, then it has a feasible integral solution z of value at least $\frac{O}{4}$. Moreover, given such an x, we may compute such a z in polynomial time.*

In fact, we show the following stronger result. Let J be any multiset of demands, and k an integer, such that for any edge $e \in T$, at most ku_e of the paths $\{Path(f) : f \in J\}$ contain e. Then J can be partitioned into $4k$ routable demand sets. We remark that in the case where all capacities are equal, finding such colourings is substantially simpler. For instance, in the case where all capacities are one, there is a $\frac{3}{2}k$ colouring, as is proved in [21]. Our result is inspired by ideas of Cheriyan, Jordan, and Ravi [13] for the half-integral case. The stronger colouring result also implies a 4-approximation algorithm on tree networks for a wavelength assignment problem in optical networks. See [17] and [1, 2] for details on the application of this result. We discuss extensions to a directed setting where each edge of T is replaced by a pair of oppositely directed arcs (with potentially different capacities). We then briefly sketch a randomized algorithm that yields a constant factor approximation. The algorithm and analysis are straight forward generalizations of the algorithms for unsplittable flow on the line [8, 9]. The constant that we can guarantee from this approach is quite large and hence we do not give a formal analysis.

Theorem 1.1 shows that (LP) has integrality gap of at most 4 for unit demands. What can we say about the integrality gap for arbitrary demands? As mentioned, Chakrabarti et al. [9] show that the integrality gap can be $\Omega(n)$ even if the tree is a path. However if $d_{\max} \leq u_{\min}$ they obtain an $O(1)$ gap for the path. Kolliopoulos and Stein [20] studied the issue of relating the integrality gap of unit demand problems to that of arbitrary demand problems in the context of a large class of packing problems. Their results are obtained by invoking their grouping and scaling technique introduced in [19] for the single source unsplittable flow problem. Since the results of [20] are presented for general and arbitrary packing problems, the bounds stated in [20] are weak (for instance, yielding integrality gaps of the order of $\sqrt{m}$). However, their technique applies equally to specific classes of packing problems where the unit demand relaxation has a better integrality gap. One such example is the single source unsplittable flow problem which spawned the essential elements in these arguments [19]. In Section 3, we develop some formalism so we can invoke grouping and scaling techniques to state tight bounds for other specific classes of combinatorial packing problems, where the unit demand relaxations have strong integrality gaps. Although our proofs essentially mimic those in [19, 20], we nevertheless believe that the statements of the theorems and the extensions we develop are convenient and useful for future work. Based on the results of Section 3, we obtain a 48-approximate algorithm for the demand flow problem on the tree with $d_{\max} \leq u_{\min}$.

Another contribution of this paper is to apply the above framework to match or improve known results for several other natural combinatorial applications. This connection between $\{0, 1\}$ packing problems and their demand versions has been missed in recent work [6, 8, 9] on these problems. In order to do this, in Section 3, we also formalize results for certain "integrality gaps with congestion". In Section 4, we apply these ideas to the line and ring networks and obtain $(2 + \epsilon)$-approximation algorithms, substantially improving the ratios provided in [9] for instance. All of these results stem from unit demand relaxations that are based on $\{0, 1\}$ constraint matrices.

Demand Flows in General Graphs: Before we proceed with our results, we make some remarks about the demand flow problem DFP in general graphs. Here we are given a supply graph $G = (V, E)$ (we may assume all capacities are 1) as well as a demand graph $H = (V, F, d)$ where for each edge $f \in F$, d_f is an integer demand. We are seeking a maximum cardinality (profit) subset $F' \subseteq F$ such that the multicommodity flow problem in G for the demands of F' is feasible. We restrict ourselves to the unit demand case, that is all d_f are 1, denoted by UNIT-DFP. A closely related problem is the maximum edge disjoint paths problem (EDP) where the flow is required to be integral (since $d_f = 1$, this corresponds to the flow being routed on a single path). We can check in polynomial time, via linear programming, whether a given set of demands can be feasibly routed if the flow is allowed to be fractional. On the other hand this same problem is NP-hard if the flow is required to be integral. EDP has been studied extensively in the literature (cf. [18]) while UNIT-DFP does not seem to have been considered before. It is known that the integrality gap for the natural LP for EDP is $\Omega(\sqrt{n})$ [15] which has recently been shown to be tight [11]. For directed graphs, Guruswami et al. [16] have shown that EDP is NP-hard to approximate to within a factor better than $\Omega(n^{\frac{1}{2}-\epsilon})$. However, for undirected graphs the problem was known only to be APX-hard [15] until recently; the same reduction applies to UNIT-DFP as well since the APX-hardness is shown on a tree. We remark that the standard instances where there is a gap for the natural linear relaxation essentially arise from a grid, where for any pair of commodity paths that cross, there is a unit capacity arc in their intersection [15, 16]. Thus, the LP has a solution of value $k/2$ (where k is the number of commodities), while at most one demand can be satisfied integrally. However, for such instances of UNIT-DFP, there is a routable subset of size $\Omega(k)$; thus the gap between the optimum values for EDP and UNIT-DFP on the same instance can be large. UNIT-DFP is an interesting problem in its own right and we believe understanding its complexity would also shed light on the complexity of EDP. In some subsequent work that was motivated by the definition of UNIT-DFP in this paper, [10] gives a poly-logarithmic approximation for UNIT-DFP (in [10] UNIT-DFP is called the all-or-nothing multicommodity flow problem). Further, for both EDP and UNIT-DFP a super-constant hardness was shown in [3]; the current best inapproximability result is that unless $NP \subseteq ZPTIME(n^{\text{polylog}(n)})$ there is no $O(\log^{1/2-\epsilon} n)$ approximation [4]. We remark that results for UNIT-DFP extend to DFP.

2 Unit Demands: Integer Multicommodity Flow on a Tree

In this section we prove Theorem 1.1; that is, for any feasible x to (LP) there is an integral solution achieving at least $\frac{1}{4}$ of x's profit. Coincidentally, we prove the following related decomposition result for multicommodity flows on a tree[1].

Theorem 2.1 *Let x be a feasible solution to (LP) where all d_e's are 1 and suppose k is an integer such that kx is an integral vector. Then there exist feasible integral solutions $z^1, z^2, \ldots, z^{4k}$ such that $kx \leq \sum_i z^i$.*

We now give a proof of Theorem 2.1. For the rest of the section, we assume without loss of generality, by possibly introducing new supply edges, that demand edges are incident only to leaves of the tree. Let k be an integer and x a feasible solution to (LP) where all demands are 1. Furthermore, suppose kx is integral, and let J be a multiset which for each demand edge f contains kx_f copies of f. We give an algorithm which partitions J into $4k$ routable subsets $R_1, R_2, \ldots, R_{4k}$.

[1] This is inspired by the integer decomposition property of a polyhedron, introduced in [7].

Note that we do not know a priori that k is polynomially bounded[2], and so this need not give rise to a polynomial time 4-approximate algorithm. However, we may adapt the arguments to find, in polynomial-time, a $\frac{1}{4}$-optimal integral solution to (LP) for an arbitrary unit demand instance; we describe this at the end of the section. Our algorithm is based on a tree colouring problem that is described below.

Binned Tree Colouring: An instance consists of an integer k, a capacitated tree (T, u), rooted at a fixed leaf node v^*, and a multiset of undirected demand edges J. For each edge $e \in T$, the number of edges of J which lie in e's fundamental cut is at most ku_e. (The *fundamental cut* of e consists of all edges with endpoints in each of the two connected components of $T - e$.) In addition, each leaf $v \neq v^*$ has a partition of the edges of J incident to v, denoted by $\delta_J(v)$, into "bins" $B_1(v), \ldots, B_{n_v}(v)$ such that $|B_i(v)| \in [1, 2k)$ for each i, and $n_v \leq u_{vp(v)}$ where $p(v)$ is the parent of v in the rooted tree. Our objective is to find a colouring of the edges of J such that each colour class J_i is a routable subset and for each leaf v, and each bin i, the edges of $B_i(v)$ all have different colours. We call such a colouring a *bin colouring* for the instance $T, u, J, \{B_i(v)\}_{i,v}$.

We now prove the following theorem which essentially implies Theorem 2.1 (we wrap up the loose details afterwards).

Theorem 2.2 *Each instance of the binned colouring problem is* $4k$*-colourable.*

Proof: Consider the tree T as "hanging down" from the root v^*. We prove the result by induction on the size of T. If T consists of a single edge vv^* then we may colour J as follows. For each bin $B_i(v)$, with edges $e_1, e_2, \ldots, e_s$ say, we colour each edge e_j with colour j. Clearly this uses at most $2k$ colours. Moreover, any colour j can occur at most once in each bin, and hence the number of edges of colour j is at most the number of bins n_v. Since $n_v \leq u_{vv^*}$, we are done.

So suppose that T has a *remote* node $v \neq v^*$, that is, a node which is not a leaf node, and is adjacent to at most one non-leaf node. Let $v_1, v_2, \ldots, v_l$ be the leaf nodes which are adjacent to v. We create a new instance as follows. Our new tree T' is obtained by contracting $\{v_1, v_2, \ldots, v_l, v\}$ to a single node which we shall refer to by v'. The set J' is obtained by the same contraction and then dropping any loop edges incident to v'. Finally, we must create the bins for our new leaf v'. Let $B_1, B_2, \ldots, B_t$ be the sub-partition of $\delta_{J'}(v')$ obtained from the bins of the v_i's by taking their intersections with the cut $\delta_J(v')$ in the shrunken graph (throwing out any empty sets). We create the bins for v' greedily as follows. First, make any of the bins of size at least k, new bins also for v'. Now for the remaining bins, start packing them together one by one until a bin of size at least k is obtained; it is designated a new bin for v (its size is clearly less than $2k$). Let x be the parent of v in T. Since we shrunk the children of v to create v', x is the parent of v' in T'. Each new bin we create for v', except perhaps the last one, has size at least k. Hence it follows that $n_{v'} \leq u_{vx} = u_{v'x}$.

Now by the induction hypothesis, we may find a bin colouring for the smaller instance obtained by the above process. We show that this induces a partial colouring of J which can be extended to a bin colouring for our original instance. First, consider some original leaf v_i and the edges $L_i = \delta_J(v_i) \cap \delta_{J'}(v')$ (which are now all coloured). Recall that these edges were originally partitioned into at most $u_{v_i v}$ bins, and each of these bins, was included in some bin of v. Thus any colour could have been assigned to at most $u_{v_i v}$ of the edges in L_i. In particular, this shows that our partial colouring does not violate any of the edge capacity constraints $u_{v_i v}$.

It remains, to complete the colouring on the edges amongst the v_i's. We now greedily extend the colouring. Suppose that e is an edge joining v_i and v_j which lies in a bin B_i for v_i, and bin B_j

[2]Using simple rounding and scaling ideas, for any fixed $\epsilon > 0$, one could alter x to another feasible vector x' such that: (i) kx' is integral, (ii) k is polynomially bounded, and (iii) $wx' \geq (1 - \epsilon)wx$.

for v_j. Call a colour *used* at B_i if some edge in B_i has already been assigned this colour. There are at most $2k-2$ such colours used since $|B_i| < 2k$. It follows that there are at least 4 colours which are not used in either B_i or B_j. Assign e one of these colours. After we complete this process we obtain a colouring which satisfies all of the bin constraints. Each colour class is a routable set of demands, since by induction, the load on any non-leaf edge is at most its capacity. And for any leaf node say v_i, the number of edges of one colour, is at most the number of bins it has, which is at most $u_{v_i v}$ by assumption. ■

We may apply this theorem to obtain Theorem 2.1 as follows. The only minor point that we must address is to make sure that none of the $4k$ routable sets contains any demand edge more than once (since (LP) has an upper bound constraint for each demand). One easily adapts the induction hypothesis to make sure this is the case. In particular, we restrict our leaf bins, so that for any demand edge $f = uv$, all copies of f occur in the same bin at u, and the same bin at v. Since there are at most k copies of any such f, one sees that this is always possible.

We mention that the Binned Tree Colouring Problem could also be defined for *bidirected trees*. These arise in directed multicommodity flow problems where the supply graph is obtained from a tree by replacing each edge e, by a pair of oppositely directed arcs, each with its own capacity. The (directed) demand graph then consists of arcs $f = (u, v)$ where there is a request for a single unit of flow to be sent along the unique directed path $P(f)$ from u to v. The *load* of a set of demands J on an arc a in a bidirected tree T, is $|\{f \in J : a \in P(f)\}|$. A set of demands J is *feasible* for a bidirected tree T with arc capacities u, if for each arc a, u_a is at least the load of J on a. Here again, one may use the same induction procedure from the proof of Theorem 2.2 to show the following.

Theorem 2.3 *Let T be a bidirected tree with integer capacities on the arcs and let u_a denote the capacity of arc a. If J is a set of directed "demand" arcs which imposes a load of at most ku_a on each arc of T, then J may be partitioned into $4k$ subsets each of which is routable on T.*

We now return to the proof of the final claim in Theorem 1.1, to find a polynomial time 4-approximate algorithm. We give the argument below. To this end, suppose that x is a basic feasible solution for (LP). We process x in a manner similar to the proof of Theorem 2.1, in order to create a routable set with at least $\frac{1}{4}$ the profit of x. As in Theorem 2.2, we prove something stronger. Namely, we again root the tree at a leaf node and think of the tree dangling downwards. In addition, each leaf has its incident demand edges partitioned into bins. Note that bins in this context do not contain multiple copies of a demand edge as in the proof of Theorem 2.2. Each bin B has the property that $x(B) = \sum_{e \in B} x_e < 2$; further, the number of bins at each leaf is at most the capacity of the edge incident to it. Clearly, we may always create such a "binning" at the leaves. We prove inductively that there exist integral feasible solutions r^i to (LP) such that

1. $r^i(B) \le 1$ for each bin B, that is, $r^i_j = 1$ for at most one $e_j \in B$ (we say at most one edge in B is "in" r^i),
2. $x = \sum_i \lambda_i r^i$ for some choice of nonnegative λ_i's with $\sum_i \lambda_i = 4$.

In fact, we show that if the size of the support of x is q, then we need at most $(q+1)$ r^i's. In particular, if x is a basic solution, we need at most $n+1$ to obtain our convex combination of the vector $\frac{x}{4}$. In this case, the combination is produced in time polynomial in n.

The base case is similar to before, we have a tree with a single edge vv^*. Suppose there are q demand edges $e_1, e_2, \ldots, e_q$, and suppose that the bins for v are $B_1, B_2, \ldots, B_t$ with $t \le u_{vv^*}$. We construct the integral solutions greedily as follows. For $i = 1, 2, \ldots$ perform the following to create a solution r^i. For $j = 1, 2, \dots, t$ if B_j is nonempty, then add exactly one edge $e_h \in B_j$, chosen arbitrarily, to r^i, i.e., set $r^i_h = 1$. Once we have looked at all t bins, we set λ_i to be $\min_{g:r^i_g=1} x_{e_g}$. We then reduce the fractional value of each edge assigned to r^i by the amount λ_i. Any edge whose x value is reduced to 0, is deleted from consideration. We repeat this process until all bins are empty. We may obviously associate each r^i to a unique demand edge which was deleted upon construction of r^i. Thus the total number of r^i's constructed in this process is at most q. Also note that $\sum_i \lambda_i \leq 2$, since at each stage i, the "size" of a maximum bin (i.e., $x(B)$ for a bin B) is reduced by λ_i. Finally, we may add a last solution r^{q+1}, say, corresponding to the empty routable set, and assign this a value $4 - \sum_i \lambda_i$.

The induction step is also similar to before. We consider contraction of a remote node which gives rise to a new leaf v'. The bins at v' are constructed from the partial bins from its descendant leaves. By induction, we obtain vectors r^i and multipliers λ_i in the smaller tree. A simple argument shows that each r^i is a feasible integral solution for the original instance as well.

We now extend this combination back to the original graph. In doing so, we must account for the demand edges joining two of the original leaves missing in the shrunken graph. That is, we must extend the combination to satisfy the bin constraints at these leaves. We can incorporate these demands one at a time, increasing the number of r^i's by at most one each time. So let $e_\ell = \alpha\beta$ be a demand edge between two leaves α and β which were shrunk in the reduction. Let B_α, B_β be the two bins which contained the edge e_ℓ. We may scan the solutions $r^1, r^2, \dots$ in order. Each time, if adding e_ℓ to the solution r^i destroys the bin condition (that is $r^i(B_\alpha) = 1$ or $r^i(B_\beta) = 1$), we move on. Otherwise, e_ℓ can be added to r^i. So we set $r^i_\ell = 1$ (add e_ℓ to that solution). Now if $x_{e_\ell} > \lambda_i$, then set $x_{e_\ell} = x_{e_\ell} - \lambda_i$ (reduce e_ℓ's demand in our "running" fractional solution). Otherwise, divide r^i into two solutions: one $r^{i,1}$ with $\lambda_{i,1} = x_{e_\ell}$, and the second copy with $r^{i,2}_\ell = 0$, and $\lambda_{i,2} = \lambda_i - x_{e_\ell}$. Note that this procedure can only increase the number of r^i's once per each demand edge e_ℓ; on the iteration where x_{e_ℓ} is finally reduced to 0. The last question is whether we could process all the r_i's without fully covering an x_{e_ℓ}. Note that e_ℓ cannot be added to some r^i only if $r^i(B) = 1$ for $B = B_\alpha$ or $B = B_\beta$; in this case we call i bad for e_ℓ. But we have that $x(B_\alpha) + x(B_\beta) < 4$, by choice. Also, $\sum_{i\ bad} \lambda_i \leq x(B_\alpha) + x(B_\beta) - x_{e_\ell} \leq 4 - x_{e_\ell}$. In other words, $\sum_{i\ not\ bad} \lambda_i \geq x_{e_\ell}$, and hence we have enough room to add e_ℓ to the convex combination.

2.1 A simple randomized algorithm

We now describe a randomized rounding algorithm that also yields a constant factor approximation. While the approximation bound is quite large, we believe its simplicity makes it of interest.

Let x be a feasible solution to (LP). We root the tree T arbitrarily and this naturally defines an ancestor-descendant relationship on the nodes of the tree. The depth of a node is its distance from the root. For a demand edge $f = uv$, let $\ell(f)$ denote the depth of the least common ancestor of u and v. We refer to $\ell(f)$ as the depth of f. The algorithm has two stages. In the first stage, we pick a set of demand edges using randomized rounding: for some sufficiently large constant $\alpha > 1$, edge f is picked independently with probability x_f/α. Let A be the collection of chosen edges. In the second stage we pick a set $B \subseteq A$ such that B is feasible. This is done as follows. We sort the demand edges in A, in *non-decreasing* order of their depth. We start with B as the empty set. We sequentially process the edges of A in the sorted order. When considering f, we add it to B, if adding f to B does not violate the feasibility of current set of edges in B. Otherwise we discard f. At the end of the algorithm, it is clear that the set of edges in B is feasible.

The above algorithm is a generalization of the algorithm for the line where the demands are sorted by their left end point. Calinescu et al. [8] developed and analyzed this algorithm for the line in the context of unsplittable flow with uniform capacities. In particular they applied this algorithm for *small* demands: a demand f is small if d_f is sufficiently small compared to that of the capacity of the line. In [9] the analysis was extended to the case of non-uniform capacities which requires some additional ideas. Here we observe that the analysis in [9] can be adapted to prove a constant factor bound on the algorithm presented above for trees, provided we restrict ourselves to unit demands and integer capacities. The analysis requires a large value for α and this yields a poor approximation ratio. We refer the interested reader to [9] for details of the analysis.

3 General Packing Problems

In this section we develop the appropriate notation to be able to assert that the integrality gap for a general "demand" problem is related by a constant factor to the unit demand version of the problem. We also extend this to certain "c-relaxed" versions which we use later to obtain improvements for some combinatorial packing problems considered in the literature.

3.1 Column-Restricted Packing Integer Programs

We consider certain classes of packing problems that arise from $\{0,1\}$ matrices as follows. In the following, let A be a $\{0,1\}$ matrix with m rows and n columns. (Generalizations to non-negative matrices are discussed at the end of the section.) For a vector $d \in \mathbf{R}^n_+$, we denote by $A[d]$ the matrix obtained from A by multiplying each entry in column i by d_i. We restrict our attention to *column-restricted packing integer programs* (CPIP), introduced by Kolliopoulos and Stein [20]. Each such problem is of the form $\max\{wx : A[d]x \leq b, x \in \{0,1\}^n\}$ for some choice of rational vectors w, d, b. We generally assume w, d, b to be integral. [3]

These column-restricted PIPs model the outcome of altering the original packing problem $\max\{wx : Ax \leq b, x \in \{0,1\}^n\}$, by adding demand values d_i to the items (columns) being packed. In generalizing their own techniques from [19], Kolliopoulos and Stein [20] devise a grouping and scaling technique to show that the integrality gap for such CPIPs are "of similar quality to those" for the $\{0,1\}$ problems. Their main objective is to establish bounds for general column-restricted packing problems. In contrast, our thrust is to examine special classes of CPIPs. As such, we first use their ideas to explicitly relate the integrality gap of column-restricted PIPs as a function of the gap for the underlying $\{0,1\}$ PIP problems; our goal here is to obtain theorem statements that are convenient and useful in applications although the proofs essentially mimic those in [19, 20]. We then indicate more general scenarios where these ideas hold. We apply these ideas to concrete packing problems: these applications have been missed by several recent papers [6, 8, 9, 24].

We now formalize some of the concepts required. For a convex body P over $\mathbf{R}^n$ and objective vector $w \in \mathbf{R}^n$, the *integrality gap* for the optimization problem $\gamma = \max\{wx : x \in P\}$ is the ratio between the fractional optimum γ and the optimal value of an integral solution, that is, $\frac{\gamma}{\max\{wx : x \in P_I\}}$. Here P_I denotes the *integer hull* of P, that is the convex hull of all integer vectors in P.

We are interested in bounding the integrality gap for classes of integer programs. Each class $\mathcal{P}$ consists of problems induced by pairs P, w where P, w lie in some fixed space $\mathbf{R}^n$. The *integrality gap* for such a class is simply the supremum of the integrality gaps for individual problems in $\mathcal{P}$.

A collection of vectors $W \subseteq \mathbf{Z}^n$ is *closed* if for any vector $w \in W$, the vector w' obtained by setting some $w_j = 0$ is also in W. In the following, for a matrix A and closed collection W we

[3] We can convert rational valued w into an integer valued w by scaling. We will also often scale d, b simultaneously.

denote by $\mathcal{P}(A,W)$ the class of problems of the form $\max\{wx : Ax \leq b, x \in [0,1]^n\}$ for some $w \in W$ and vector $b \in \mathbf{Z}_+^m$. We then denote by $\mathcal{P}^{dem}(A,W)$ the class of problems of the form $\max\{wx : A[d]x \leq b, x \in [0,1]^n\}$ for some $w \in W$, and vectors $b \in \mathbf{Z}^m, d \in \mathbf{Z}_+^n$ with $d_{max} \leq b_{min}$. The notion of closedness is motivated by the fact that the integrality gap for a restricted class of weight vectors can be superior to that for the class of all weight vectors. In order to apply the grouping and scaling techniques of [19, 20], the only requirement on the class of weight vectors, is that it be closed.

We then have the following result whose proof follows precisely the lines of analysis used by Kolliopoulos and Stein in their study of single source unsplittable flow [19] and CPIPs [20]. We obtain a slightly better constant than the factor of $1/0.075 \simeq 13.333$ reported in [19], due to a small error in their final calculation.

Theorem 3.1 *Let A be a $\{0,1\}$ matrix and W be a closed collection of vectors. If the integrality gap for the collection of problems $\mathcal{P}(A,W)$ is at most Γ, then the integrality gap for the collection of problems $\mathcal{P}^{dem}(A,W)$ is at most $11.542\Gamma \leq 12\Gamma$.*

One may note that with suitable definitions, a variant of the above result can shown for general non-egative integer matrices A. We discuss this at the end of this section.

To prove Theorem 3.1 we set up some notation. Let $\Pi(A,b,w)$ denote the packing problem $\max\{wx : Ax \leq b, x \in [0,1]^n\}$ from for some w, b and let $\Pi(A,b,d,w)$ be a problem of the form $\max\{wx : A[d]x \leq b, x \in [0,1]^n\}$ for some d. (While we do not require d, b to be integer in these definitions, this only arises in uniform demand instances where the vector b is an integer multiple of the demand in question.) Given a subset S of $\{1, 2, \ldots, n\}$, we denote by A^S the matrix A restricted to the columns in S. Given S, we have two naturally defined new problems $\Pi(A^S, b, w^S)$ and $\Pi(A^S, b, d^S, w^S)$. Given a fractional solution x for $\Pi(A,b,w)$, its restriction to S is denoted by x^S.

In the following we use two positive parameters $\alpha, \beta \in (0,1)$. We optimize these parameters at the end to obtain the best ratio. It is useful to have the setting $\alpha = \beta = 1/3$ in mind to follow the proof. We call a demand f *large* if $d_f \geq \beta d_{\max}$, otherwise we call it *small*. We show how to obtain two types of integral solutions: one obtained only using the large demands, and the other by selecting only from the small demands. The better of the two solutions will give the desired approximation. We first work with the large demands, where we may obtain at least $\frac{\beta}{2\Gamma}$ times the fractional profit (under x) of these demands.

Lemma 3.2 *Let S be the set of large demands. Given a fractional solution x to $\Pi(A,b,d,w)$ there is an integral solution to $\Pi(A^S, b, d^S, w^S)$ of value at least $\frac{\beta}{2\Gamma}\sum_{f \in S} w_f x_f$.*

Proof: Given x, S and $\Pi(A^S, b, d^S, w^S)$ we create, as follows, a new instance $\Pi(A^S, c, \hat{d}, w^S)$ and a feasible fractional solution y^S to it. First $\hat{d}$ denotes the vector in $\mathbf{R}^S$ with each component equal to d_{max}; thus we are setting all the demands to be $d_{\max}$. We set $y^S = \frac{\beta}{2} x^S$. For $1 \leq j \leq m$, we set $c_j = \lfloor b_j/d_{\max} \rfloor d_{\max}$, in other words we set c_j to be the largest integer multiple of $d_{\max}$ not exceeding b_j. We observe two easy facts. First, the solution y^S is feasible for the instance $\Pi(A^S, c, \hat{d}, w^S)$. Here we use the fact that the demands are large; we increase all demands to $d_{\max}$ which requires that we multiply the fractional solution by a factor of β to maintain feasibility. The additional factor of $1/2$ is to ensure that the fractional solution is feasible even after we reduce the capacities to be integer multiples of $d_{\max}$. Second, any feasible integral solution to $\Pi(A^S, c, \hat{d}, w^S)$ translates into a feasible solution to $\Pi(A^S, b, d^S, w^S)$ of the same value. These two observations combined with the fact that $\Pi(A^S, c, \hat{d}, w^S)$ is a uniform demand instance with c an integer multiple of $\hat{d}$, and hence has an integrality gap of at most Γ, yields the lemma. ■

It is the above lemma where we need the stronger condition that $d_{max} \leq b_{min}$ as opposed to the weaker, natural condition of $A_{ij}d_j \leq b_i$. This is because we have no control over which columns may end up receiving demands of size βd_{max}, and such demands need to be rounded up to d_{max}.

Now we address the small demands.

Lemma 3.3 *Let S be the set of small demands. Given a fractional solution x to $\Pi(A, b, d, w)$ there is an integral solution to $\Pi(A^S, b, d^S, w^S)$ of value at least $\alpha(1 - \frac{\beta}{1-\alpha})\frac{1}{\Gamma}\sum_{f\in S} w_f x_f$.*

Proof: For $t \geq 0$, let S_t be the subset of small demands f such that $d_f \in (\alpha^{t+1}\beta d_{\max}, \alpha^t \beta d_{\max}]$. For each t we construct a new instance $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$ and a feasible fractional solution y^t in S_t-space, as follows. For $f \in S_t$, we set $y^t_f = \alpha(1 - \frac{\beta}{1-\alpha})x_f$ and we set $d^t_f = \alpha^t\beta d_{\max}$. We define the *load* (under x) on constraint i from demands in S_t in x, denoted by ℓ^t_i as $\sum_{f\in S_t} A_{if}d_f x_f$. We set c^t_i to be the smallest integer multiple of $\alpha^t\beta d_{\max}$ larger than $(1 - \frac{\beta}{1-\alpha})\ell^t_i$. In particular, $c^t_i \leq (1 - \frac{\beta}{1-\alpha})\ell^t_i + \alpha^t\beta d_{\max}$.

By construction $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$ is a uniform demand problem. It is easily verified that y^t is a feasible solution for this instance. Hence, by our assumption on the integrality gap of the $\{0, 1\}$ instances, there exists an integral solution z^t to $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$ of value at least $\frac{1}{\Gamma}\sum_{f\in S_t} w_f y^t_f = \frac{1}{\Gamma}\alpha(1 - \frac{\beta}{1-\alpha})\sum_{f\in S_t} w_f x_f$.

We now argue that combining the solutions z^t into one single solution z gives a feasible integral solution to $\Pi(A^S, b, d^S, w^S)$ of value at least $\frac{1}{\Gamma}\alpha(1 - \frac{\beta}{1-\alpha})\sum_{f\in S} w_f x_f$. From the analysis in the previous paragraph, the value of z is at least as much as we claim. We show that z is feasible. Consider an arbitrary constraint i: since z^t is feasible for $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$ it follows that $\sum_{f\in S_t} A_{if}d^t_f z^t_f \leq c^t_i$. By construction, for $f \in S_t$, $d_f \leq d^t_f = \alpha^t\beta d_{\max}$, therefore we have that

$$\sum_{f\in S_t} A_{if}d_f z^t_f \leq c^t_i \leq (1 - \frac{\beta}{1-\alpha})\ell^t_i + \alpha^t\beta d_{\max}.$$

Hence the load on constraint i in the combined solution z is at most $\sum_{t\geq 0}((1 - \frac{\beta}{1-\alpha})\ell^t_i + \alpha^t\beta d_{\max})$ which is at most $(1 - \frac{\beta}{1-\alpha})\sum_t \ell^t_i + \frac{\beta}{1-\alpha}d_{\max}$. By the feasibility of x, $\sum_t \ell^t_i \leq b_i$. We also have that $d_{\max} \leq b_i$, therefore the load on constraint i due to z is at most b_i. This shows that z is a feasible integral solution. ■

In the proof of the above lemma we scale down the fractional solution x by $\alpha(1-\frac{\beta}{1-\alpha})$. The factor α accounts for rounding *up* the demands in S_t to $\alpha^t\beta d_{\max}$. The factor $(1 - \frac{\beta}{1-\alpha})$ is to make room for the additional capacity to the tune of $\alpha^t\beta d_{\max}$ that we add to each instance $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$ to make the capacities integral multiples of $\alpha^t\beta d_{\max}$.

Proof of Theorem 3.1: Let L denote the set of large demands and S denote the set of small demands. From Lemma 3.2 we obtain an integer solution of value at least $\frac{\beta}{2\Gamma}\sum_{f\in L} w_f x_f$. From Lemma 3.3 we obtain an integer solution of value least $\alpha(1 - \frac{\beta}{1-\alpha})\frac{1}{\Gamma}\sum_{f\in S} w_f x_f$. For a given β it is easy to verify that the expression $\alpha(1 - \frac{\beta}{1-\alpha})$ is maximized when $\alpha = 1 - \sqrt{\beta}$, hence for small demands we obtain an integer solution of value at least $(1 - 2\sqrt{\beta} + \beta)\frac{1}{\Gamma}\sum_{f\in S} w_f x_f$.

Let $\epsilon \in [0, 1]$ be defined by the equation $\sum_{f\in S} w_f x_f = \epsilon\sum_f w_f x_f$, in other words ϵ is the fraction of the total weight of small demands in the fractional solution. From the above analysis we are guaranteed an integral solution of value $\max\{\epsilon(1-2\sqrt{\beta}+\beta), (1-\epsilon)\beta/2\}\frac{1}{\Gamma}\sum_f w_f x_f$. The algorithm can choose β to maximize this expression but has no control over the distribution of ϵ. Hence we can bound the integrality gap by the expression $\max_{\beta<1} \min_{0\leq\epsilon\leq 1}\{\frac{1}{\Gamma}\epsilon(1-2\sqrt{\beta}+\beta), \frac{1}{\Gamma}(1-\epsilon)\beta/2\}$. Numerical computation shows that this expression is at least $\frac{1}{11.542\Gamma}$. Hence the integrality gap is at most 11.542Γ. Setting $\alpha = \beta = 1/3$ yields a simple analysis that shows that the integrality gap is at most 12Γ. ■

We now give useful corollaries to Theorem 3.1. First, we justify the natural expectation that the integrality gap should tend to 1 as $d_{\max}/b_{\min} \to 0$. We denote by $\mathcal{P}^{\epsilon dem}(A,W)$ the class of column-restricted packing problems such that $d_{\max} \leq \epsilon b_{\min}$. In the corollaries below we choose ϵ sufficiently small to obtain a clean expression for the dependence of the integrality gap as a function of ϵ.

Corollary 3.4 *Let A be a $\{0,1\}$ matrix and W be a closed collection of vectors. If the integrality gap for the collection of problems $\mathcal{P}(A,W)$ is at most Γ, then the integrality gap for the collection of problems $\mathcal{P}^{\epsilon dem}(A,W)$ for $\epsilon < (3-\sqrt{5})/2$, is at most $\frac{1+\sqrt{\epsilon}}{1-\sqrt{\epsilon}-\epsilon}\Gamma$ (upper bounded by $(1+O(1)\sqrt{\epsilon})\Gamma$). This holds even under the weaker condition that $\max_j A_{ij}d_j \leq \epsilon b_i$, for each $i = 1,2,\ldots,m$.*

Proof: This follows from Lemma 3.3 with $\beta = \epsilon$ and $\alpha = \frac{1}{1+\sqrt{\epsilon}}$. An examination of the proof of Lemma 3.3 shows that the lemma holds even if we are only guaranteed that $\max_j A_{ij}d_j \leq \beta b_i$, for each $i = 1,2,\ldots,m$. ■

There are examples of packing problems (for instance ring routing [23]) for which relaxing the capacity constraints by an additive constant independent of the input parameters yields an improved integrality gap. For a $\{0,1\}$ packing problem $\max\{wx : Ax \leq b, x \in \{0,1\}^n\}$, constant c, the *c-relaxed* integrality gap is Γ if the value of the optimum solution to the relaxed problem $\max\{wx : Ax \leq b + \hat{c}, x \in \{0,1\}^n\}$ is at least $1/\Gamma$ times the value of the (fractional) solution to $\max\{wx : Ax \leq b, x \in [0,1]^n\}$, where $\hat{c}$ denotes the m-vector with all components equal to c.

Corollary 3.5 *Let A be a $\{0,1\}$ matrix and W be a closed collection of vectors. If the c-relaxed integrality gap for the collection of problems $\mathcal{P}(A,W)$ is at most Γ, then the integrality gap for the collection of problems $\mathcal{P}^{\epsilon dem}(A,W)$ for $\epsilon < \frac{1}{4(1+c)^2}$, is at most $\frac{1+\sqrt{\epsilon}}{1-(1+c)(\sqrt{\epsilon}+\epsilon)}\Gamma$ (upper bounded by $(1+O(1)(1+c)\sqrt{\epsilon})\Gamma$). This holds even under the weaker condition that $\max_j A_{ij}d_j \leq \epsilon b_i$, for each $i = 1,2,\ldots,m$.*

Proof: The proof follows closely the proof of Lemma 3.3. In Lemma 3.3 the fractional solution is scaled down by a factor of $\alpha(1-\frac{\beta}{1-\alpha})$. As we remarked, the factor $(1-\frac{\beta}{1-\alpha})$ is to make room for the extra capacity we add to the sub-problems generated by demands in S_t. Since we work with c-relaxed integrality gap, we need to add an additional capacity of $c\alpha^t\beta d_{\max}$ to the subproblem t. Hence we need a scaling factor of $\alpha(1-\frac{(1+c)\beta}{1-\alpha})$ to accommodate this extra space. By choosing $\alpha = \frac{1}{1+\sqrt{\epsilon}}$ and $\beta = \epsilon$ we get the desired result. ■

In the next corollary, we show that one may dispense with the condition $d_{max} \leq b_{min}$ by explicitly allowing some congestion in each constraint.

Corollary 3.6 *Let A be a $\{0,1\}$ matrix and W be a closed collection of vectors. If the integrality gap for the collection of problems $\mathcal{P}(A,W)$ is at most Γ, then for every b and $w \in W$, and for every $\beta > 1$ there is an integral solution to $\max\{wx : A[d]x \leq \beta b + \frac{\beta}{\beta-1}d_{\max}\}$ of value at least $1/\Gamma$ times the value of the optimum fractional solution to $\max\{wx : A[d]x \leq b\}$.*

Proof: The proof follows along the lines of that for Lemma 3.3, however, we allow the capacities to be violated but do not scale down the solution x. Let $\alpha = 1/\beta$. For $t \geq 0$, let S_t be the set of demands f such that $d_f \in (\alpha^{t+1}d_{\max}, \alpha^t d_{\max}]$. We create a new instance $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$ as follows. For $f \in S_t$, we set $d^t_f = \alpha^t d_{\max}$. We define the *load* on constraint i from demands in S_t in x, denoted by ℓ^t_i as $\sum_{f \in S_t} A_{if} d_f x_f$. We set c^t_i to be the smallest integer multiple of $\alpha^t d_{\max}$ larger than ℓ^t_i/α. Note that $c^t_i \leq \ell^t_i/\alpha + \alpha^t d_{\max}$. We observe that the fractional solution x^{S_t} is feasible for $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$.

As before we obtain integral solutions z^t for each of the above instances and combine them to obtain a solution z. Since we did not scale down the fractional solution and the integrality gap of each of the subproblems is at most Γ, the value of z is at least $1/\Gamma$ times the value of $\sum_f w_f x_f$. It remains to show that for $i \in 1, 2, \ldots, m$, $\sum_f A_{if} d_f z_f \leq \beta b_i + \frac{\beta}{\beta-1} d_{\max}$.

The solution z^t satisfies the capacity constraints for $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$. Hence it follows that z satisfies the capacity constraints defined by $\sum_t c^t$ which by construction is dominated by $\beta b + \frac{\beta}{\beta-1} d_{\max}$. ■

Algorithmic aspects: Theorem 3.1 and its corollaries have been phrased in terms of integrality gaps. However, it is straightforward to see that one can obtain algorithmic versions of the theorems. We give an informal description of the algorithmic version using the notation developed earlier. Given a fractional solution y for a problem $\{\max wx : A[d]x \leq b\}$ where w belongs to a closed collection of vectors W, we can find in polynomial time an integer solution y' such that $wy' \geq wy/11.542\Gamma$ and $A[d]y' \leq b$ provided we have access to the following oracle: given a fractional solution z to $\{\max w'x : Ax \leq b'\}$ for any $w' \in W$ and integer vector b', the oracle outputs an integer solution z' such that $wz' \geq wz/\Gamma$ and $Az' \leq b'$. We note that in some cases the number of columns of A might be exponential in the size of an underlying problem. This happens naturally in the setting of routing problems when one uses a path formulation. It might be possible to obtain, in polynomial time, a fractional solution to the problem in which the number of non-zero variables is polynomially bounded. One can still apply the grouping and scaling ideas in this setting as long as we have the desired oracle as mentioned above.

Nonnegative matrices: Our focus so far has been on packing problems induced by $\{0, 1\}$ matrices. We now consider arbitrary non-negative integral matrices and discuss the extent to which the grouping and scaling ideas and Theorem 3.1 can be generalized. As before, for a given non-negative matrix A we are interested in packing problems of the form $\max\{wx : x \in P(A, b)\}$. Here $P(A, b) = conv\{x \in \{0, 1\}^n : Ax \leq b\}$. For a matrix A let $A_i = \max_{j=1}^n A_{ij}$ denote the maximum entry in row i of A. For a given matrix A and a closed collection of vectors W we define, as before, two classes of problems $\mathcal{P}(A, W)$ and $\mathcal{P}^{dem}(A, W)$. $\mathcal{P}(A, W)$ is the class of problems of the form $\max\{wx : Ax \leq b, x \in [0, 1]^n\}$ for some $w \in W$ and vector $b \in \mathbf{Z}^m_+$ such that $A_i \leq b_i$ for $1 \leq i \leq m$. This added restriction on the set of b's that we allow is necessary to avoid an integrality gap of ∞ (when A is $\{0, 1\}$ this was enforced implicitly by requiring b to be an integer vector). $\mathcal{P}^{dem}(A, W)$ denotes the class of problems of the form $\max\{wx : A[d]x \leq b, x \in [0, 1]^n\}$ for some $w \in W$, and vectors $b \in \mathbf{Z}^m, d \in \mathbf{Z}^n_+$ with $A_i d_{\max} \leq b_i$ for $1 \leq i \leq m$. Note that for $\{0, 1\}$ matrices this amounts to our requirement that $d_{\max} \leq b_{\min}$. With the above notation in place we can state the following.

Theorem 3.7 *Let A be a nonnegative integer matrix and W be a closed collection of integer vectors. If the integrality gap for the collection of problems $\mathcal{P}(A, W)$ is at most Γ, then the integrality gap for the collection of problems $\mathcal{P}^{dem}(A, W)$ is at most* 11.542Γ.

The proof of the above theorem follows by adapting the proof of Theorem 3.1. The essential difference is in the proof of Lemma 3.3. When considering demands in S_t we set c^t_i to be the smallest integer multiple of $A_i \alpha^t \beta d_{\max}$ larger than $(1 - \frac{\beta}{1-\alpha})\ell^t_i$. The extra factor of A_i ensures that in the resulting uniform demand instance $\Pi(A^{S_t}, c^t, d^t, w^{S_t})$, we have the property $A_i^{S_t} d^t_{\max} \leq c^t_i$; this allows us to bound its integrality gap by Γ.

4 Applications to Combinatorial Demand Problems

We now discuss some applications of results in the previous section to combinatorial problems.

4.1 Demand Flow in a Tree

Our original task was to show that the natural LP formulation for the multicommodity demand flow problem has an $O(1)$ integrality gap for instances where $d_{max} \leq u_{min}$ and the supply graph is a tree. Theorems 1.1 and 3.1 now imply that the integrality gap is indeed at most 48. Moreover, we find in polynomial time, an integral solution delivering at least $\frac{1}{48}$ times the profit of the optimal fractional solution.

4.1.1 Demand Flow in a Line

When the supply graph is a line (path), the demand problem has been studied for its application to resource allocation [6, 8, 9]. In [8] a $2+\epsilon$-approximation is provided for the uniform capacity problem improving upon the 3-approximation in [6]. The main observation in [8] is that when $d_{\max} \leq \epsilon U$ where U is the common capacity of the edges, the integrality gap of the LP is $1/(1-O(\sqrt{\epsilon \ln 1/\epsilon}))$: this is proved by an interesting use of randomized rounding with alteration. In [9] this approach is extended to the non-uniform capacity case and an $O(1)$ approximation is presented. We improve the approximation ratio for the non-uniform capacity case to $(2+\epsilon)$ as follows.

We follow the broad outline developed in [9]. For a given demand f, we define its *bottleneck* capacity $b(f)$ as the smallest capacity edge on Path(f). We call a demand f *δ-large* if $d(f) \geq \delta \cdot b(f)$, otherwise it is *$\delta$-small*. We make use of the following lemma from [9].

Lemma 4.1 ([9]) *If $d_{\max} \leq u_{\min}$, in a feasible integral solution, the number of δ-large demands that cross any edge is at most $2\lfloor 1/\delta^2 \rfloor$.*

Using the above lemma it is shown in [9] that an optimal solution to an instance in which all demands are δ-large can be found via dynamic programming in $O(n^{O(1/\delta^2)})$ time. For δ-small demands, using randomized rounding with alteration, an $O(1)$-approximation was shown. We can improve this substantially by applying Corollary 3.4. Note that the underlying $\{0,1\}$ packing problem has a totally unimodular matrix and hence its integrality gap is 1. Therefore, from Corollary 3.4, the integrality gap of the LP for the problem on δ-small demands when $\delta < \frac{3-\sqrt{5}}{2}$ is at most $\frac{1+\sqrt{\delta}}{1-\sqrt{\delta}-\delta}$ which is $(1+O(1)\sqrt{\delta})$. To obtain a $(2+\epsilon)$-approximation we partition the demands into δ-large and δ-small demands, for a choice of δ specified later. We obtain an optimal solution to the instance with only the δ-large demands in $O(n^{O(1/\delta^2)})$ time. For δ-small demands we solve the linear programming relaxation and use Corollary 3.4 to obtain a $(1+O(1)\sqrt{\delta})$-approximation. (In contrast the bound shown in [8] for δ-small demands is $(1+O(1)\sqrt{\delta \log 1/\delta})$ even for the uniform capacity case.) We then choose the better of the two solutions. It is easy to see that this yields a $(2+O(1)\sqrt{\delta})$-approximation since either the δ-large demands or the δ-small demands have at least half the value of an optimal solution. By choosing δ to be $O(\epsilon^2)$ we obtain the desired $(2+\epsilon)$-approximation. The running time of the algorithm is dominated by the time to solve the δ-large demands.

Theorem 4.2 *For the demand flow problem on the line with $d_{\max} \leq u_{\min}$, there is a $(2+\epsilon)$-approximation algorithm that runs in time $O(n^{O(1/\epsilon^4)})$.*

We close this subsection by mentioning that the problem on the line is just a special case of directed path packing in oriented trees. Here we have a tree $T = (V, A)$ where each arc has an integer capacity u_a. We also have some "demand" arcs F, such that each $a = (u, v) \in F$ creates a directed cycle when added to T. Each a also has a weight w_a. A subset $S \subseteq F$ is a *path packing*, if for each arc $a \in T$, at most u_a of the arcs of S have ends in distinct components of $T - a$. Finding a maximum weight path-packing problem has a natural formulation given by a network matrix (hence totally unimodular). Therefore this formulation has an integrality gap of 1. The demand version of the path packing problem thus has a 12-approximation via Theorem 3.1. Note that flows on the line just arise by taking T to be a directed path.

4.2 Unsplittable Demand Flow in a Ring

We now consider the demand flow problem in a ring. The ring is the simplest network where in addition to deciding which demands to route, we also need to specify the route. We restrict ourselves to the *unsplittable* flow problem on the ring, that is if we satisfy a demand f, we route the entire demand d_f along one of the two paths available for f. It is shown in [9] that an α-approximation for the demand flow problem on the line can be used to obtain an $(\alpha + 1)$-approximation for the unsplittable flow problem on the ring. From the $(2 + \epsilon)$-approximation for the line that we described above, we obtain a $(3 + \epsilon)$-approximation for the ring. Below we describe an improved $(2 + \epsilon)$-approximation.

As in the case of the line we would like to classify demands as δ-large and δ-small. We say that a demand f is δ-large for an edge e if $d_f \geq \delta u_e$, otherwise it is δ-small for e. However, since each demand has a choice of two routes, it is possible that a demand is large on one of the routes and small on the other. For a demand f let $R_1(f)$ and $R_2(f)$ denote the two possible routings. We say a demand is *routed as δ-large* if f is routed along $R_i(f)$ for $i \in \{1, 2\}$ and $d_f \geq \delta \min_{e \in R_i(f)} u_e$. Otherwise it is *routed as δ-small*. Consider a fixed optimal solution OPT and let L be the set of demands in OPT that are routed as δ-large and S be the set of demands that are routed as δ-small. We find two solutions, L' and S' such that $w(L') \geq w(L)$ and $w(S') \geq (1 - \epsilon/4)w(S)$ and take the better of the two to obtain the desired approximation ratio.

Let a be a minimum capacity edge on the ring, that is $u_a = u_{\min}$. We first guess the set of demands in L that use a in OPT. Since every demand in L is routed as δ-large, there are at most $\lfloor 1/\delta \rfloor$ demands in L that use a. Hence there are at most $O(n^{\lfloor 1/\delta \rfloor})$ guesses. Let L_a be the right guess. If $f \notin L_a$ then, in subsequent steps of the algorithm, we prohibit f from being routed on a as δ-large. We remove a from the ring to obtain a line network where the capacities of the edges are reduced by the amounts used up by L_a. It is easy to see that every demand in $L - L_a$ is routed as δ-large in the resulting line network. We can use dynamic programming as we did in Section 4.1.1 to obtain an optimal solution to the problem of finding the largest weight subset of δ-large demands that can be routed in the line network obtained as above. Note that $L - L_a$ is a feasible solution to this problem. It follows that $L' = L_a \cup A$, where A is the solution obtained by the dynamic programming, satisfies the property that $w(L') \geq w(L)$. Given L_a, the running time for the dynamic programming, as we saw earlier, is $O(n^{O(1/\delta^2)})$.

Now we describe how to find S' given L_a. Let S'' be the set of demands such that f is δ-small on the route for f that does not use a. Clearly every demand in S belongs to $S'' \setminus L_a$. We classify S'' into two types of demands - tiny and medium. A demand f is *tiny* if $d_f < \delta u_{\min}$, otherwise it is *medium*. Note that tiny demands can be routed as δ-small in either direction while medium demands can be routed as δ-small only on the route that does not use a. We consider the following integer program for demands in S''. For each demand f there are two variables x_f^1 and x_f^2 corresponding to the two possible routings $R_1(f)$ and $R_2(f)$. In the integer program we prohibit medium demands from using the route that contains a.

$$
\begin{array}{rcll}
\max \sum_{f \in S''} w_f x_f & & \text{s.t} & \\
x_f^1 + x_f^2 & = & x_f & \forall f \in S'' \\
\sum_{f: e \in R_1(f)} d_f x_f^1 & \leq & u_e & \forall e \in E \\
\sum_{f: e \in R_2(f)} d_f x_f^2 & \leq & u_e & \forall e \in E \\
x_f^i & = & 0 & \forall f \in S'', f \text{ medium}, i \in \{1,2\}, a \in R_i(f) \\
x_f, x_f^1, x_f^2 & \in & \{0,1\} & \forall f \in S''.
\end{array}
$$

The integer program above models a slightly restricted version of the the following constrained demand flow problem on rings. We are given a capacitated ring network and demands that need to be routed unsplittably. In addition, for each demand f, it is specified whether it can routed on $R_1(f)$, $R_2(f)$ or on either. The goal is to find the largest profit subset of demands that can be routed feasibly.

Clearly S identifies a feasible solution to the above integer program. We now solve the linear relaxation for this problem. We then round the LP to obtain an integral solution that is within $(1 - O(\sqrt{\delta}))$ of the LP value using Corollary 3.5. However, we need to establish that the problem modeled by the above integer program has a c-relaxed integrality gap for some small fixed constant c. We do that now.

Lemma 4.3 *For the* $\{0,1\}$ *version of the constrained demand flow problem in rings, the* 5.5*-relaxed integrality gap of the natural linear relaxation is* 1.

Proof: Consider a solution x to the linear program. We say that a demand f is integrally routed if $x_f, x_f^1, x_f^2 \in \{0,1\}$. We can ignore integrally routed demands by removing them from the demand set and reducing the capacities of edges appropriately. We call a demand *split* if it has positive amount of flow in both clockwise and counterclockwise directions on the ring. Note that only tiny demands can be split. Let S be the set of split demands in the solution x. We first show that if we increase each edge capacity by 2.5, we may find a solution of equal or greater value in which each demand in S is integrally routed. We accomplish this as follows. First suppose there are two split demand edges ij, kl that do not cross, i.e., the nodes appear in clockwise order as i, j, k, l say. Then we may decrease flow on ij's (kl's) clockwise path starting from j (resp. l), and increase it on its clockwise path starting from i (resp. k). This does not increase the load on any edge. Thus we may assume that any pair of split demand edges are crossing. We say that a demand f is fulfilled if $x_f = 1$. We claim that at most one split demand is not fulfilled; for otherwise, we may transfer some ϵ amount of flow for the less profitable demand, to the more profitable one. If there is a split unfulfilled demand, we may fulfil it by increasing the load of all edges by at most 1/2. Let $x_s(e)$ be the load on e by the split demands. Thus these demands can be fractionally routed on the ring with capacities $\lceil x_s(e) \rceil$. Thus by a result from [23] we may route all split demands (note that are all fulfilled) integrally by increasing the load of each edge by at most by 1. The total increase in the edge capacities is at most 2.5, 1/2 for making one demand fulfilled, 1 for making the capacities $x_s(e)$ integral, and 1 to apply the result from [23]. We can now ignore these integrally routed demands.

Note that for the remaining demands, either $x_f^1 = 0$ or $x_f^2 = 0$. Consider an arbitrary edge α of the ring and partition the set of flow paths for x into two classes, Y_1, Y_2 where Y_1 is the set of paths that do not contain α and Y_2 is the rest. Consider our fractional solution x to the problem. Let $x_i(e)$ denote the capacity of edge e that is used up by demands in Y_i. We first define a capacity function $u_1(e) = \lceil x_1(e) \rceil$. Call I_1 the instance obtained by restricting to paths of Y_1 and capacity vector u_1 on the line obtained by deleting α. Since the natural linear formulation for this instance, is given by a totally unimodular matrix, we may obtain an integral solution yielding at least as much profit as x when restricted to Y_1.

Now consider a demand $f \in Y_2$ and let Q denote the path on which f routes its flow. We may traverse Q in the clockwise direction to yield a decomposition of the form $Q = L, \alpha, R$. We call L and R the *left* and *right* parts of Q. For each $e \neq \alpha$, we think of two copies, one for carrying flow from left parts, and one for carrying flow for right parts. For each e we define $x_l(e)$ and $x_r(e)$ to be the load induced on e by the left or right paths respectively. We then define $u_2(\alpha) = \lceil x_2(\alpha) \rceil$ and for $e \neq \alpha$, we define $u_2(e) = \lceil x_l(e) \rceil + \lceil x_r(e) \rceil \leq x_2(e) + 2$. By considering two copies of each e, we can essentially split the ring into a path, and so may solve an instance, I_2, of maximum integer flow on the line again, using capacities $\lceil x_l(e) \rceil, \lceil x_r(e) \rceil$. See Fig 2 for an illustration of splitting the ring into a path. Hence we find an integral packing of paths in Y_2 that gives as much profit as x restricted to Y_2. One sees that these paths now pack into the original ring with capacities u_2.

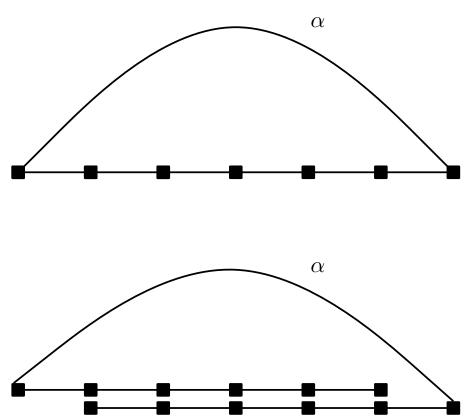

Figure 2: Splitting a ring into a line for demands using the edge α.

Combining the solutions for I_1, I_2 we obtain an integral solution with as much profit as x. Moreover, the paths pack into capacity vector $u_1(e) + u_2(e) \leq x_1(e) + x_2(e) + 3$. For getting rid of the split demands, for edge e we used capacity $x_s(e) + 2.5$. Hence the solution packs into a ring with capacities $u(e) + 5.5$. Therefore the 5.5-relaxed integrality gap for the problem is 1. ∎

From Lemma 4.3 and Corollary 3.5, it follows that the integrality gap for the linear relaxation for tiny and medium demands is $(1 + O(1)\sqrt{\delta})$. Choosing $\delta = O(\epsilon^2)$ yields a solution S' such that $w(S') \geq (1 - \epsilon/4)w(S)$. Combining this with the optimal algorithm for L yields the following theorem.

Theorem 4.4 *There is a $(2 + \epsilon)$-approximation algorithm for the unsplittable flow problem on the ring for instances with $d_{\max} \leq u_{\min}$ that runs in $O(n^{O(1/\epsilon^4)})$ time.*

4.3 General Unsplittable Flows

In applications to unsplittable flows, one needs a minor refinement of Theorem 3.1. Let A be a $\{0,1\}$ matrix. Let $\mathcal{V} = V_1 \cup V_2 \cup \ldots V_t$ be a partition of the n column variables and let $w \in \mathbf{Z}^t$. We consider the $\{0,1\}$ packing problem $\{\max \sum_{i=1}^{t} w_i \cdot \min(1, \sum_{j \in V_i} x_j) : Ax \leq b, x \in \{0,1\}^n\}$. We consider a closed collection of vectors $W \subseteq \mathbf{Z}^t$. Given a $\{0,1\}$ matrix A, a partition V of the column variables, and a closed collection of vectors $W \subseteq \mathbf{Z}^t$, we obtain the class of problems $\mathcal{P}(A, \mathcal{V}, W)$. We denote by $\mathcal{P}^{dem}(A, \mathcal{V}, W)$ the class of column-restricted packing problems in $\mathcal{P}^{dem}(A, W)$ arising from demand vectors d with the property that $d_i = d_j$ if $i, j \in V_q$ for some q. The previous proofs extend in a straightforward fashion to yield:

Theorem 4.5 *Let A be a $\{0,1\}$ matrix, $\mathcal{V}$ a t-partition of its columns and W be a closed collection of vectors $\mathbf{Z}^t$. If the integrality gap for the collection of problems $\mathcal{P}(A, \mathcal{V}, W)$ is at most Γ, then the integrality gap for the collection of problems $\mathcal{P}^{dem}(A, \mathcal{V}, W)$ is at most 11.542Γ.*

We note that Corollaries 3.4 and 3.5 also generalize to the above setting. In the above formulation, let z_i denote the sum $\sum_{j \in V_i} x_i$. An integral solution corresponds to $x \in \{0,1\}^n$. The objective function ensures that attention can be restricted to solutions where in addition $z \in \{0,1\}^t$. Motivated by applications to DFP, we also consider a relaxed problem by asking for solutions that allow $x \in [0,1]^n$ while requiring that $z \in \{0,1\}^t$. Once again, it is straightforward to extend the gap results to this setting.

In applications to unsplittable flow, we may choose to associate each V_i with the flow path variables for a fixed commodity i. The objective function ensures that solutions do not route a commodity multiple times. In applications to DFP, we require that $z \in \{0,1\}^t$ but allow $x \in [0,1]^n$.

We mention that the best approximation ratio known for the maximum edge-disjoint paths problem in general undirected graphs is $O(\sqrt{n})$ [11]. For directed graphs it is known to be $\Omega(n^{1/2-\epsilon})$-hard, for any $\epsilon > 0$. However, there are several classes of graphs for which better approximation ratios are known for EDP, and so grouping and scaling leads to improved bounds for the associated unsplittable flow problem as well. One notable case is the single-source problem for which the maxflow-mincut theorem characterizes solvability, and hence an integrality gap of one for the natural path-formulation for EDP. In this case, one obtains the result of Kolliopoulos and Stein [20]: it establishes a gap of 11.542 for the single-source unsplittable flow, subsequently improved [14] to 4.425 using different techniques. Another case where an $O(1)$ approximation is possible is when G is planar. In [12], it is shown that given a feasible fractional routing for an instance of integer multicommodity flow where each edge has capacity at least 4, one may obtain a set of integrally routable demands whose size is a constant fraction of the original total fractional flow. Using this, one may also obtain a constant fraction of the total possible unsplittable flow, if in addition we can increase each edge capacity by $O(d_{max})$. In a similar fashion, the poly-logarithmic bounds for UNIT-DFP [10] extend to DFP.

5 Conclusions

We conclude with several open problems. For the tree with unit demands, the main open problem is resolving the 3/2-conjecture of Cheriyan et al. [13]. Indeed, this may be provable by showing that an approximate 3/2-integer decomposition property holds, such as we showed a 4-approximate decomposition in Section 2. Interestingly, this would be a generalization of Shannon's edge colouring theorem for multigraphs (by considering the restriction to the star). It would also be interesting to obtain the smallest c such that the c-relaxed integrality gap for the problem is 1. We are unable to prove this for any constant c. A better understanding of basic solutions seems necessary for answering the above two questions. For arbitrary demands, we believe that a direct approach to the tree problem will enable the ratio of 48 to be improved substantially. In some recent work [5] a quasi-polynomial time approximation scheme is obtained for the demand flow problem on the line and ring even when the condition $d_{\max} \le u_{\min}$ is not satisfied.

Another direction of interest is the following. Theorem 3.1 shows that the integrality gaps for (suitably constrained) demand packing problems are within a constant factor of the gap for corresponding relaxation for their unit demand versions. We have not offered advice on what to do if the original formulation for the unit demand version suffers from a large integrality gap. One natural approach is to work on tightening the original formulation for the unit demand problem by adding new valid inequalities, or *cutting planes*. In doing so, however, even if the matrix A was originally $\{0, 1\}$, the new inequalities may not be. Thus we must work in the setting where we have a general nonnegative integer matrix A. In this context can we show that cutting planes derived for the unit-demand matrix A be extended in a natural fashion to the corresponding demand packing problems? The hope is that this would enable us to prove a theorem akin to Theorem 3.1 where the integrality gap Γ is now for the strengthened (with cutting planes) unit-demand problem.

Acknowledgements: The major portion of this work was done during a summer visit of the second author to Bell Laboratories. We are grateful to ONR for supporting this visit (N00014-02-M-0125). We thank the anonymous reviewers for their comments and corrections.

References

[1] M. Andrews and L. Zhang. Bounds on fiber minimization in optical networks with fixed fiber capacity. *Proc. of IEEE INFOCOM*, 409–419, 2005.

[2] M. Andrews and L. Zhang. Complexity of wavelength assignment in optical network optimization. *Proc. of IEEE INFOCOM*, 2006.

[3] M. Andrews and L. Zhang. Hardness of the undirected edge-disjoint paths problem. *Proc. of* 37^{th} *ACM STOC*, 276–283, 2005.

[4] M. Andrews, J. Chuzhoy, S. Khanna and L. Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. *Proc. of* 46^{th} *IEEE FOCS*, 226–244, 2005.

[5] N. Bansal, A. Chakrabarti, A. Epstein and B. Schieber. A Quasi-PTAS for Unsplittable Flow on Line Graphs. *Proc. of 38th ACM STOC*, 721–729, 2006.

[6] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor and B. Schieber. A unified approach to approximating resource allocation and scheduling. *JACM*, 48(5), 1069–90, 2001. Preliminary version in *Proc. of* 32^{nd} *ACM STOC*, 2000.

[7] S. Baum and L. E. Trotter Jr. Integer rounding and polyhedral decomposition for totally unimodular systems. *Optimization and Operations Research* (Proceedings Workshop Bad Honnef, 1977; R. Henn, B. Korte, W. Oettli, eds.) [Lecture Notes in Economics and Mathematical Systems 157], Springer, Berlin, 1978, 15–23.

[8] G. Calinescu, A. Chakrabarti, H. Karloff and Y. Rabani. Improved Approximation Algorithms for Resource Allocation. *Proc. of* 9^{th} *IPCO*, 401–414, Springer-Verlag LNCS, 2002.

[9] A. Chakrabarti, C. Chekuri, A. Gupta and A. Kumar. Approximation Algorithms for the Unsplittable Flow Problem. To appear in *Algorithmica*. Preliminary version in *Proc. of* 5^{th} *APPROX*, 51–66, 2002.

[10] C. Chekuri, S. Khanna and F. B. Shepherd. The All-or-Nothing Multicommodity Flow Problem. *Proc. of* 36^{th} *ACM STOC*, 156–165, 2004.

[11] C. Chekuri, S. Khanna and F. B. Shepherd. An $O(\sqrt{n})$ Approximation and Integrality Gap for Disjoint Paths and Unsplittable Flow. *Theory of Computing*, 2(7), 137–146, 2006.

[12] C. Chekuri, S. Khanna and F. B. Shepherd. Edge-disjoint paths in planar graphs with constant congestion. *Proc. of* 38^{th} *ACM STOC*, 757–766, 2006.

[13] J. Cheriyan, T. Jordan and R. Ravi. On 2-coverings and 2-packings of laminar families. *Proc. of* 7^{th} *ESA*, 510–520, LNCS, Springer-Verlag, 1999.

[14] Y. Dinitz, N. Garg and M. X. Goemans. On the single source unsplittable flow problem, *Combinatorica* 19:1–25, 1999. Preliminary version in *Proc. of IEEE FOCS*, 1998.

[15] N. Garg, V. Vazirani and M. Yannakakis. Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees. *Algorithmica*, 18(1):3-20, 1997. Preliminary version in *Proc. of* 20^{th} *ICALP*, 1993.

[16] 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, *Journal of Computer and System Sciences*, 67(3):473–496, 2003. Preliminary version in *Proc. of* 29^{th} *ACM STOC*, 1999.

[17] T. Erlebach, A. Pagourtzis, K. Potika and S. Stefanakos. Resource allocation problems in Multifiber WDM Tree Networks. *Proc. of Workshop on Graph Theoretic Concepts in Computer Science*, LNCS 2880, Springer-Verlag, 218–229, 2003.

[18] J. Kleinberg. *Approximation algorithms for disjoint paths problems.* Ph.D Thesis, MIT, 1996.

[19] S. G. Kolliopoulos and C. Stein. Approximation Algorithms for Single-Source Unsplittable Flow, *SIAM J. Computing* (31), 919–946, 2002. Preliminary version in *Proc. of* 38^{th} *IEEE FOCS*, 1997.

[20] S. G. Kolliopoulos and C. Stein. Approximating Disjoint-Path Problems using Packing Integer Programs. *Mathematical Programming A*, (99), 63–87, 2004. Preliminary version in *Proc. of* 6^{th} *IPCO*, 1998.

[21] P. Raghavan and E. Upfal. Efficient routing in all-optical networks. *Proc. of* 26^{th} *ACM STOC*, 134–143, 1994.

[22] A. Schrijver. Theory of Linear and Integer Programming. John Wiley and Sons, 1986.

[23] F. B. Shepherd and L. Zhang. An Augmentation Algorithm for Mincost Multicommodity Flow on a Ring, *Discrete Applied Mathematics*, (110):301–315, 2001.

[24] F. B. Shepherd, A. Vetta. The Demand Matching Problem. To appear in *Math. of Operations Research*. Preliminary version in *Proc. of* 9^{th} *IPCO*, 457–474, 2002.

[25] L. Wolsey, private communication, Oberwolfach, 2002.