Single–Source Unsplittable Flow

Jon M. Kleinberg*

Abstract

The max-flow min-cut theorem of Ford and Fulkerson is based on an even more foundational result, namely Menger's theorem on graph connectivity. Menger's theorem provides a good characterization for the following *single–source disjoint paths problem*: given a graph G, with a *source vertex* s and *terminals* $t_1, \ldots, t_k$, decide whether there exist edge–disjoint s-t_i paths, for $i = 1, \ldots, k$.

We consider a natural, NP-hard generalization of this problem, which we call the *single–source unsplittable flow problem*. We are given a source and terminals as before; but now each terminal t_i has a *demand* $\rho_i \leq 1$, and each edge e of G has a capacity $c_e \geq 1$. The problem is to decide whether one can choose a single s-t_i path, for each i, so that the resulting set of paths respects the *capacity constraints* — the total amount of demand routed across any edge e must be bounded by the capacity c_e.

The main results of this paper are constant–factor approximation algorithms for three natural optimization versions of this problem, in arbitrary directed and undirected graphs. The development of these algorithms requires a number of new techniques for rounding fractional solutions to network flow problems; for two of the three problems we consider, there were no previous techniques capable of providing an approximation in the general case, and for the third, the randomized rounding algorithm of Raghavan and Thompson provides a logarithmic approximation. Our techniques are also of interest from the perspective of a family of NP-hard load balancing and machine scheduling problems that can be reduced to the single–source unsplittable flow problem.

Our basic approximation algorithms can be used to provide a constant–factor approximation for the general unsplittable flow problem, without the single–source restriction, on the family of unit–capacity, *densely embedded graphs* introduced by the author and Éva Tardos. Our work also presents an interesting parallel with recent work of Schrijver, Seymour, and Winkler, which deals with "unsplittability" constraints in the context of SONET ring routing.

*Laboratory for Computer Science, MIT, Cambridge MA 02139. Author supported by an ONR Graduate Fellowship. Present address: Department of Computer Science, Cornell University, Ithaca NY 14853. Email: kleinber@cs.cornell.edu. On leave at IBM Almaden Research Center, San Jose CA 95120.

1 Introduction

Issues related to admission control and routing in high–speed networks have led to recent algorithmic work on the classical *disjoint paths problem.* We are given a graph G, and pairs of vertices that wish to communicate over paths through G. The problem is to decide whether all the given pairs can be connected by mutually edge–disjoint paths in G. If this is possible, we say that the set of pairs is *routable.*

This is a basic problem in combinatorial optimization and algorithmic graph theory. A special case of it is the focus of Menger's foundational theorem on graph connectivity [19], generalized into the max-flow min-cut theorem of Ford and Fulkerson [7, 8]. It is one of Karp's original NP-complete problems [13]; and there has since been a large amount of work done to characterize exactly solvable special cases of the problem. (See e.g. the surveys by Frank [9], Schrijver [26], and Robertson and Seymour [25].)

The relation to routing problems in networks is quite direct. A communication network is modeled as a graph (either directed or undirected), and each link has some bounded *capacity.* Certain pairs of physically separated nodes in the network, which we will call *terminal pairs*, wish to establish a connection on which they can transmit data at a given *rate.* Now, in most models of virtual–circuit routing (see e.g. [21, 10, 4]), bandwidth for each connection must be explicitly reserved on a single path; thus, our goal is to find a path for each terminal pair, so that the cumulative rate of transmission crossing any link does not exceed the link's capacity. Again, we will say that the set of terminal pairs is *routable* if these paths can be found.

The disjoint paths problem captures this general network routing problem in quite a crude way; it assumes that all links have the same capacity, and that every connection requires all the capacity on each link that it uses. And while this simplification retains much of the intractability that arises from *routing* issues, it obscures a second, independent, source of intractability that arises in network routing problems — namely, *packing* issues.

To illustrate what we mean, consider a graph G that simply consists of two nodes, s and t, with m parallel links between them, each of capacity 1. Suppose we are given k terminal pairs; the i^{th} has one end at s, the other at t, and requires a transmission rate of ρ_i. Then although the "routing" problem in G is trivial — G has only two nodes — deciding whether the given set of terminal pairs is *routable* in G is still NP-complete; it is equivalent to a bin–packing problem with m bins of capacity 1, and weights $\rho_1, \ldots, \rho_k$. (Bin-packing, with two bins, is another of Karp's original NP-complete problems [13].)

Motivated by these considerations, one can formulate the *unsplittable flow problem* (UFP), a generalization of the disjoint paths problem. We say that a *routing* of $\mathcal{T}$ in G is a choice of an s_i-t_i path P_i for each $(s_i, t_i) \in \mathcal{T}$.

> Given a graph G (directed or undirected) with a *capacity* $c_e \geq 1$ on each edge e,
> a set $\mathcal{T}$ of k terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$, and
> a *demand* $\rho_i \leq 1$ for each terminal pair (s_i, t_i),
> decide whether there exists a routing of $\mathcal{T}$ in G, with paths $\{P_i\}$, such that the following *capacity constraint* is met for each edge e:
>
> $$\sum_{i:e \in P_i} \rho_i \leq c_e.$$

That is, we require ρ_i units of *flow* to be sent from s_i to t_i, on a single path, and the cumulative amount of flow sent across any edge cannot exceed its capacity. This problem captures both of the sources of intractability mentioned above — it contains the disjoint paths problem when all capacities and demands are equal to 1, and it trivially contains the bin–packing problem by the reduction described above.

In the literature on the disjoint paths problem and its applications, one can identify three main optimization versions of the problem that have been extensively studied [23, 22, 11, 2, 15, 16, 3, 24, 1, 14]. Adapting these to the setting of the unsplittable flow problem, let us state them as follows.

(1) **Maximization.** Find a routable subset of the terminals of maximum total demand. We denote this maximum by $\alpha(\mathcal{T})$.

(2) **Routing in Rounds.** Partition the terminals into as few "rounds" of communication as possible, where the set of terminals assigned to each round must be routable. We denote this minimum by $\chi(\mathcal{T})$.

(3) **Congestion and Relative Congestion.** We define the *congestion* of a routing of $\mathcal{T}$ in G to be the maximum amount of flow crossing any edge e of G. We define the *relative congestion* of a routing to be the maximum, over all edges e, of the total amount of flow crossing e divided by the edge capacity c_e. Let $\nu(\mathcal{T})$ denote the minimum congestion of any unsplittable routing of $\mathcal{T}$ in G, and $\nu_0(\mathcal{T})$ the minimum relative congestion.

Single–Source Unsplittable Flow

Developing non–trivial approximation algorithms for optimization problems (1) and (2) above, in arbitrary graphs, appears to be beyond the reach of current techniques even for the basic disjoint paths problem. Approximations for (1) and (2), in the case of disjoint paths, have been obtained for restricted classes of graphs [11, 24, 2, 15, 16]. Of course, designing approximations for the more general unsplittable flow problem can only be more difficult. (We note that for problem (3), the general *randomized rounding* technique of Raghavan and Thompson [23] provides a logarithmic approximation to $\nu(\mathcal{T})$ in any graph, and this extends to the general setting of unsplittable flow.)

In this paper, we develop approximation algorithms, in arbitrary graphs, for the *single–source unsplittable flow problem.* This is the case of the problem in which all terminal pairs share a common *source* vertex; so $s_i = s$, for some vertex s and all $(s_i, t_i) \in \mathcal{T}$. By the bin–packing reduction described above, the single–source UFP is NP-complete even on a two–node graph. The single–source UFP is tightly connected with several foundational problems in combinatorial optimization — it is a generalization of the *single–source disjoint paths problem*, which is the subject of Menger's theorem [19], and its fractional relaxation (when the flow from s to each t_i can be split over several paths) is simply the *maximum flow problem.* As such, we are dealing with a very basic, NP-hard, variant of maximum flow.

However, the only approximation technique currently available for this problem is the general algorithm of Raghavan and Thompson; in an arbitrary graph $G = (V, E)$, this provides an $O(\log |E|)$–approximation for $\nu(\mathcal{T})$; and it is not able to provide any non–trivial performance guarantee for $\alpha(\mathcal{T})$ or $\chi(\mathcal{T})$ unless every capacity exceeds every demand by a factor of at least $\Omega(\log |E|)$. Thus, there are no existing techniques for approximating problems (1) and (2) in the general single–source case.

Our Results

The main results of this paper are constant–factor approximation algorithms for the single–source unsplittable flow problem in arbitrary directed and undirected graphs, for each of the objective functions α, χ, and ν.

Our approximation ratios scale well with the maximum demand value. Recall that all capacities are at least 1, and let $\rho^* = \max_i \rho_i \leq 1$. In addition to our general constant–factor approximations, we show the following.

(i) There is a polynomial–time algorithm producing a routable set of total demand at least $1 - O(\sqrt{\rho^*})$ times $\alpha(\mathcal{T})$.

(ii) There is an absolute constant $\rho_0 < 1$ and a polynomial–time algorithm with the following guarantee. If $\mathcal{T}$ is routable, and $\rho^* < \rho_0$, then the algorithm will route $\mathcal{T}$ in two rounds.

(iii) There is a polynomial–time algorithm that produces a routing of all terminals in $\mathcal{T}$ with congestion at most $(1 + O(\sqrt{\rho^*/\nu(\mathcal{T})})) \cdot \nu(\mathcal{T})$.

For the sake of presentation in this version of the paper, we will not always present the tightest possible constants inside the $O()$ notation; our main goal is to show that constant–factor approximations are obtainable for these problems, and we wish to develop this as cleanly as possible. However, none of the constants that we obtain in our proofs will be very large in any case.

There are a number of points worth noting about our results, as well as some extensions.

• As a direct problem in network routing, the single–source unsplittable flow problem is situated analogously to the maximum flow problem: it is relevant to problems of point–to–point traffic, as well as more general problems in which paths must be established from many points to a "central" node in a network. As such it is a basic problem, for example, in the setting of multicast communication.

But it turns out that we can also use our single–source algorithm to attack the general unsplittable flow problem, with arbitrary sources and sinks, in the class of *densely embedded graphs* introduced by the author and Éva Tardos [16]. The algorithm we developed there [16] decomposes the graph into *clusters*, and runs a maximum flow algorithm within each cluster; and the bottleneck in adapting this algorithm to the case of arbitrary unsplittable demands was the lack of a suitable substitute for these maximum flow algorithms. Thus, using the results of this paper, we can obtain a constant–factor approximation for the general unsplittable flow maximization problem in any uniform–capacity densely embedded graph. We will describe this in Section 8.

• Our algorithms are based on new rounding methods for converting *fractional flows* into unsplittable flows. We will introduce fractional flows in the following subsection; we observed earlier that the fractional version of the single–source UFP is precisely the maximum flow problem. Our rounding techniques are fundamentally different from the randomized rounding method of Raghavan and Thompson [23]. As we mentioned above, randomized rounding only provides an $O(\log |E|)$–approximation for the minimum congestion in our case, and it cannot be applied at all for α and χ unless $\rho^* \leq 1/\log|E|$.

A line of work related to ours was introduced by Cosares and Saniee [5] in the context of SONET ring routing. The problem considered there is the unsplittable flow problem for arbitrary sources and sinks on the n-node cycle. Working from a fractional flow solution, they provide a 2-approximation for the minimum congestion of an unsplittable routing. This bound was subsequently strengthened in very recent work of Schrijver, Seymour, and Winkler [27], who were able to parametrize the performance guarantee for the minimum congestion in terms of the maximum demand ρ^*. Our basic approach is quite different from the approach taken in their work; we are able to use the fact that all terminal pairs have a common source, but must deal with an arbitrary underlying graph rather than just the cycle. What the present work shares with these papers is the emphasis on analyzing the packing constraints that arise when one imposes "unsplittability" requirements on a set of terminal pairs with varying amounts of demand.

- The single–source UFP is also valuable as a unifying problem within the context of approximation. We note that much of the appeal of the max-flow min-cut theorem has nothing to do with issue of routing in networks, but rather lies in the ease with which one can reduce a wide range of problems in combinatorial optimization to it. It turns out that this notion of reduction is also relevant in the setting of the single–source UFP for directed graphs. In addition to simple variants of the bin–packing and minimum–makespan problems, one can reduce to the single–source UFP a variety of more complex NP-hard load balancing and machine scheduling problems, analogous to those considered by Lenstra, Shmoys, and Tardos [18] and others. We will discuss this further in Section 6.

Definitions and Background

We now introduce some terminology and notation that will be used throughout the paper, and then introduce the notion of fractional flow at a precise level. For most of the paper, we will be working with single–source problems, which we will specify by a triple $(G, \mathcal{D}, s)$, where $G = (V, E)$ is the underlying graph, $\mathcal{D} \subset V$ is the set $\{t_1, \ldots, t_k\}$ of *terminals*, and $s \in V$ is the *source*. By abuse of notation, we will sometimes use $\mathcal{D}$ to refer to the set of pairs $\{(s, t_i)\}$ as well; the meaning will be clear from context. We will sometimes use $\alpha(G, \mathcal{D}, s)$ to denote the value of $\alpha(\{(s, t_i) : t_i \in \mathcal{D}\})$; and we will use analogous notation for the functions χ, ν, and ν_0. For now, we will work with undirected graphs; later we will discuss the extensions to directed graphs.

We define a fractional flow from s to $\mathcal{D}$ as follows. For each $t_i \in \mathcal{D}$ we choose
(i) a set of s-t_i paths $P_i^1, \ldots, P_i^{q_i}$, and
(ii) associated non-negative weights $y_i^1, \ldots, y_i^{q_i} \in \mathbf{R}$ such that $z_i = \sum_j y_i^j \leq \rho_i$.

Of the ρ_i units of flow from s to t_i, a total of y_i^j is routed on path P_i^j (and so $\rho_i - z_i$ units of flow are not routed at all). We say that such a flow is a *fractional routing* if $z_i = \rho_i$ for each $i = 1, \ldots, k$; that is, the total demand for each terminal pair is routed on the associated flow paths. We define the *congestion* and *relative congestion* of a fractional flow by analogy with the definitions for unsplittable flow; let $\nu_0^f(G, \mathcal{D}, s)$ denote the minimum congestion of any fractional routing from s to $\mathcal{D}$. We say that $(G, \mathcal{D}, s)$ is *fractionally feasible* if $\nu_0^f(G, \mathcal{D}, s) \leq 1$; i.e. if it has a fractional routing that satisfies the capacity constraints.

A *maximum fractional flow* (or simply a *maximum flow*) is a fractional flow that routes the maximum amount of demand subject to the capacity constraints. We use $\alpha^f(G, \mathcal{D}, s)$ to denote the amount of demand routed by a maximum flow.

Now, for a set X of vertices, let $\delta(X)$ denote the set of edges with one end in X and the other end in $V \setminus X$. Define $c(X)$ to be the sum of all capacities of edges in $\delta(X)$, and $\rho(X)$ to be the sum of all demands of terminals in $\mathcal{D} \cap X$. The *cut condition* is the following criterion: For all $X \subset V \setminus \{s\}$, we have $c(X) \geq \rho(X)$. Clearly it is a necessary condition for the existence of a fractional routing from s to $\mathcal{D}$. For our purposes, we can state the max-flow min-cut theorem of Ford and Fulkerson [7, 8] as follows.

Theorem 1.1 (Ford–Fulkerson) *Let G be an arbitrary undirected graph, s a source vertex, and $\mathcal{D}$ a set of terminals with demands.*

(i) There is a fractional routing from s to $\mathcal{D}$ if and only if the cut condition holds.

(ii) A maximum fractional flow can be found in polynomial time.

(iii) If all demands ρ_i are equal to a common value ρ, and all capacities are multiples of ρ, then there is a maximum fractional flow that is an unsplittable flow. Moreover, this unsplittable flow can be found in polynomial time.

We will be making extensive use of this theorem in constructing unsplittable flows for arbitrary sets of demands.

Techniques

Two of the main methods used to prove the max-flow min-cut theorem are *augmentation* — iteratively construct a maximum flow by increasing flow on some edges and re-routing across others — and *induction* — find a cut whose incident edges are saturated, contract the side of the cut that does not contain the source, and proceed by induction. We mention, without going into much detail, that it seems very difficult to define suitable "approximate" analogues of these methods in the setting of unsplittable flow. There are a number of basic reasons for this — in particular, both methods rely very heavily on the fact that in constructing a fractional flow, it is enough to maintain knowledge only of the amount of flow f_e crossing each edge e. In our setting, on the other hand, one must keep track of the *set of demand values* ρ_i crossing each edge; thus, for example, it matters whether $f_e = 1$ because e is carrying one unsplittable unit of flow, or because it is carrying two unsplittable half–units of flow. In terms of other methods, a straightforward randomized rounding of the fractional flow, as in [23], is well–known to incur an unavoidable logarithmic blow–up in the amount of flow crossing some edge, in very simple examples. Finally, it seems that an approach based on setting all demands to the same value, invoking Theorem 1.1(iii), and then restoring all demands to their original values is inherently incapable of providing a constant–factor approximation, since it introduces terms that depend on the ratio between the largest and smallest demands in $\mathcal{D}$.

Our basic method in this paper is closest to the last of these ideas — we reduce the problem to one in which all demands are approximately the same by a *grouping* technique. Specifically, we cover the graph G with small trees in such a way that (a) no edge is contained in too many trees, and (b) the amount of demand contained in each tree is approximately the same. We then build the unsplittable flow in two parts: we first use Theorem 1.1(iii) to find unsplittable flow paths for designated *leader nodes* in each tree; and we concatenate these flow paths with paths connecting the terminals in $\mathcal{D}$ to the leaders of their trees.

It turns out that in the undirected case, essentially any such "tree cover" of the graph that satisfies a few simple conditions is sufficient to make this work. Things are much more complicated in the directed case. Although there is a directed analogue of Theorem 1.1 to work with, one must choose the tree cover very carefully now to make sure that the source has a flow of large enough value to the leaders of the resulting trees. It is this careful construction of a "good" tree cover that will require most of the work in the algorithm for directed graphs.

The remainder of the paper is organized as follows. In Section 2, we introduce the definitions and basic lemmas involved in the construction of tree covers for G. In Sections 3, 4, and 5, we use this construction to obtain our constant–factor approximation algorithms for the three objective functions. In Section 6, we describe how to extend the algorithms to the case of directed graphs. In Section 7, we consider a minimum-cost version of the single–source unsplittable flow problem; we show that given a set of terminals that is fractionally routable in an undirected graph with cost ξ, there exists an unsplittable routing of these terminals with relative congestion at most $1+O(\sqrt{\rho^*})$ and cost at most $(3+O(\sqrt{\rho^*})\xi$. Finally, in Section 8, we describe how to use the algorithms of Section 3 to obtain a constant–factor approximation for the general unsplittable flow maximization problem in any uniform–capacity densely embedded graph.

2 Coalitions

Recall our assumption that $\max_i \rho_i \leq 1 \leq \min_e c_e$. We define $\rho^* = \max_i \rho_i \leq 1$.

By a *tree in* G, we mean a subset of the vertices of G that induces a connected subgraph, together with a spanning tree for these vertices. If T is a tree, in which each node $v \in T$ has an associated non–negative *weight* ω_v, we define a *center node* of (T, ω) to be a node $u \in T$ such that $T \setminus \{u\}$ has no component X for which

$$\sum_{v \in X} \omega_v > \frac{1}{2} \sum_{v \in T} \omega_v.$$

It is well–known that every weighted tree contains a center node.

We now introduce the notion of a *coalition*, which will serve as our way of grouping demand in G. A coalition will be a set of trees, no more than two of which contain any edge, with each terminal assigned to some tree that contains it.

Definition 2.1 *A* (weak) coalition *in* $(G, \mathcal{D}, s)$ *is a pair* $(\{T_1, \ldots, T_r\}, \tau)$*, where each* T_i *is a tree in* G*,* $\tau : \mathcal{D} \to \{T_1, \ldots, T_r\}$ *is a function, and the following properties are satisfed.*

(i) The trees are half–disjoint, *by which we mean that each edge of G belongs to at most two of the T_i.*

(ii) For each $i = 1, \ldots, k$, $t_i \in \tau(t_i)$.

We will be particularly interested in coalitions for which the total demand assigned to each tree is approximately the same:

Definition 2.2 *A* strong coalition *in $(G, \mathcal{D}, s)$ of order σ is a pair $(\{T_1, \ldots, T_r\}, \tau)$, such that the following properties are satisfed.*

(i) $(\{T_1, \ldots, T_r\}, \tau)$ is a coalition in G.

(ii) For each $j = 1, \ldots, r$, either $s \in T_j$ or we have

$$\sigma \leq \sum_{i:\tau(t_i)=T_j} \rho_i \leq \sigma + \rho^*. \tag{1}$$

The trees of a strong coalition for which inequality (1) does not hold will be called the *exceptional trees.* Note that all exceptional trees, by property (iii), contain s.

First we show that strong coalitions always exist.

Lemma 2.3 *For every $\sigma \geq 0$, there exists a strong coalition in $(G, \mathcal{D}, s)$ of order σ.*

Proof. Fix a spanning tree T of G, and perform a depth-first search of T beginning at s. Let C denote the sequence of edges (with repetitions) encountered during the depth-first search; note that C contains each edge of T exactly twice, and no other edges of G. Let $t_{\pi(1)}, \ldots, t_{\pi(k)}$ denote the order in which the terminals are first encountered while following C, where π is a permutation on $\{1, \ldots, k\}$. We partition the sequence $t_{\pi(1)}, \ldots, t_{\pi(k)}$ into r intervals, such that the sum of the values of ρ_{i_j} in each interval but the last is between σ and $\sigma + \rho^*$. This is possible since each ρ_{i_j} is at most ρ^*. Let $\mathcal{D}_j$ denote the set of terminals in the j^{th} such interval, and let C_j denote the set of edges of C that were traversed between the first visit to a vertex in $\mathcal{D}_j$ and the last visit to a vertex in $\mathcal{D}_j$. In the case $j = r$, define C_r to be the set of edges of C that were traversed between the first visit to a vertex in $\mathcal{D}_r$ and the end of the depth-first search (at s).

Now, C_j contains the edge set of a spanning tree on a superset of $\mathcal{D}_j$; let T_j be such a spanning tree. Moreover, since $G[C_r]$ contains s, we choose T_r so that it contains s. Define the function $\tau : \mathcal{D} \rightarrow \{T_1, \ldots, T_r\}$ by setting $\tau(t_i) = T_j$ if $t_i \in \mathcal{D}_j$.

We claim first that $(\{T_1, \ldots, T_r\}, \tau)$ is a coalition. Property (ii) of Definition 2.1 is immediate from the construction; to see (i), we observe that each edge e of G appears only twice in C; so it appears in at most two of the C_j and thus in at most two of the trees T_j. Now, the total demand assigned to each tree by τ is between σ and $\sigma + \rho^*$, except possibly for the tree T_r, which contains s. Thus $(\{T_1, \ldots, T_r\}, \tau)$ is a strong coalition of order σ. ∎

If Γ is a (weak) coalition in $(G, \mathcal{D}, s)$, with trees $T_1, \ldots, T_r$, we define $U(\Gamma)$ to be any set of vertices $\{u_1, \ldots, u_r\}$ such that

(i) if T_i is not exceptional, then u_i is a center node of T_i, and

(ii) if T_i is exceptional, then $u_i = s$.

We will refer to the node $u_i \in U(\Gamma)$ as the *leader* of the tree T_i. u_i will be called *exceptional* if the associated tree T_i is exceptional, and *non–exceptional* otherwise. We define a single–source UFP $(G, U(\Gamma), s)$ by associating with each $u_i \in U(\Gamma)$ any demand that does not exceed the total demand assigned by Γ to the tree T_i. We now show that one can route this set of nodes $U(\Gamma)$ without much more relative congestion than was possible for the original set of terminals.

Lemma 2.4 *Let Γ be a coalition in $(G, \mathcal{D}, s)$, and suppose that $(G, \mathcal{D}, s)$ has a feasible fractional routing. Consider the problem $(G, U(\Gamma), s)$ as defined in the previous paragraph, and suppose that the maximum demand associated with any $u_i \in U(\Gamma)$ is at most ε. Then $(G, U(\Gamma), s)$ has a fractional routing of relative congestion at most $1 + \varepsilon$.*

Proof. The feasible fractional routing for $\mathcal{D}$ in G assigns to each $t_j \in \mathcal{D}$ a set of t_j-s paths $M_j^1, \ldots, M_j^{q_j}$ and non-negative weights $y_j^1, \ldots, y_j^{q_j}$ such that $\sum_\ell y_j^\ell = \rho_j$. We use these paths to define a fractional flow for $U(\Gamma)$ in G.

For $u_i \in U(\Gamma)$, let ε_i denote the amount of demand associated with u_i in the problem $(G, U(\Gamma), s)$, and let ε_i' denote the total amount of demand associated with terminals of $\mathcal{D}$ that are assigned to the tree T_i:

$$\varepsilon_i' = \sum_{j:\tau(t_j)=T_i} \rho_j.$$

Recall that by the definition of the problem $(G, U(\Gamma), s)$, we have $\varepsilon_i \leq \varepsilon_i'$. Now if $u_i \in U(\Gamma)$ is the leader of an exceptional tree, then $u_i = s$ and hence we do not need to construct any paths at all. Otherwise, for each $t_j \in \tau^{-1}(T_i)$, let N_{ij} denote the unique u_i-t_j path in T_i. We now define the fractional flow paths for u_i as follows. For those j such that $t_j \in \tau^{-1}(T_i)$, and $\ell = 1, \ldots, q_j$, we define P_{ij}^ℓ to be the concatenation of N_{ij} and M_j^ℓ. The set of paths for u_i is then

$$\{P_{ij}^\ell : t_j \in \tau^{-1}(T_i),\ 1 \leq \ell \leq q_j\}.$$

We set the weight associated with the path P_{ij}^ℓ to be

$$y_j^\ell \cdot (\varepsilon_i / \varepsilon_i') \leq y_j^\ell. \tag{2}$$

We define the *M-part* of a path P_{ij}^ℓ to be the portion consisting of edges of M_j^ℓ, and the *N-part* of P_{ij}^ℓ to be the portion consisting of edges of N_{ij}.

We must verify that this set of flow paths, for all non–exceptional u_i, has relative congestion at most $1 + \varepsilon$. First observe that the set of all M-parts of these flow paths contributes a relative congestion of at most 1: these M-parts are in one-to-one correspondence with the original (feasible) fractional flow paths for $(G, \mathcal{D}, s)$, and the weight assigned to each M-part is less than or equal to the weight assigned to the corresponding fractional flow path for $(G, \mathcal{D}, s)$, by Equation 2. The set of all N-parts of these paths contributes a congestion of at most ε to each edge. To see this, observe that since each non–expectional u_i is a center node of its associated tree, the total demand across any edge in G due to N-parts of paths for u_i is at most $\frac{1}{2}\varepsilon_i \leq \frac{1}{2}\varepsilon$. Since no edge of G is contained in more than two trees of Γ, the total demand across any edge in G due to the N-parts of all flow paths is at most ε. Since all edges in G have capacity at least 1, it follows that the total *relative* congestion due to the

N-parts of all paths is also at most ε. Adding the bounds for the two parts of each path, one obtains the statement of the lemma. ∎

One can also prove this lemma by multiplying the capacities in G by $1+\varepsilon$ and then verifying the cut condition. We thank Michel Goemans for suggesting the proof given above.

3 Congestion

In this section, we use strong coalitions to obtain three related results, in the single–source case: a bound on the ratio between the optimum relative congestion for fractional and unsplittable flows, an approximate max-flow min-cut theorem for unsplittable flow, and an approximation algorithm for the minimum congestion of an unsplittable flow.

Theorem 3.1 *Let $(G, \mathcal{D}, s)$ be fractionally feasible. Then it has an unsplittable routing of relative congestion at most* $1 + 4\sqrt{\rho^*} + 4\rho^* = (1 + 2\sqrt{\rho^*})^2$.

Proof. We set $\sigma_0 = \frac{1}{2}\sqrt{\rho^*} \geq \frac{1}{2}\rho^*$, and let Γ be a strong coalition in $(G, \mathcal{D}, s)$ of order σ_0. As before, choose a set of nodes $U(\Gamma)$. To each non–exceptional node $u_i \in U(\Gamma)$, we associate a demand of σ_0; to each exceptional u_i we associate a demand of 0.

Let G' denote the graph obtained from G by multiplying all capacies by $1+\sigma_0$, and then rounding each capacity up to the nearest multiple of σ_0. Note that each resulting capacity c'_e satisfies

$$c'_e \leq (1+\sigma_0)c_e + \sigma_0 \leq (1+2\sigma_0)c_e.$$

Now, since $(G, \mathcal{D}, s)$ is assumed to be fractionally feasible, it follows from Lemma 2.4 that $(G', U(\Gamma), s)$ is fractionally feasible. But all demands in $(G', U(\Gamma), s)$ are equal to 0 or σ_0, and all capacities are multiples of σ_0, so by Theorem 1.1 there is a feasible unsplittable flow for $(G', U(\Gamma), s)$.

Let Q_i, for $i = 1, \ldots, r$, denote the u_i-s path associated with this feasible unsplittable flow. We now describe the construction of the unsplittable flow in $(G, \mathcal{D}, t)$, by specifying a path for each terminal t_i. So let $t_j \in \mathcal{D}$, and suppose $\tau(t_j) = T_i$. We define R_j to be the unique t_j-u_i path in T_i. We then define P_j, the flow path for t_j, to be the concatenation of R_j and Q_i. We define the *Q-part* of P_j to be the portion consisting of edges of Q_i, and the *R-part* of P_j to be the portion consisting of edges of R_j.

It remains only to compute the relative congestion of the flow defined by the paths $\{P_j\}$. The above arguments showed that if each Q_i were to carry at most σ_0 units of flow, then the relative congestion on edge e due to the Q-parts of all flow paths in G would be at most $1 + 2\sigma_0$. But by part (ii) of Definition 2.2, each Q_i may carry up to $\sigma_0 + \rho^*$ units of flow, and thus the relative congestion on e due the Q-parts of all paths is at most

$$\frac{\sigma_0 + \rho^*}{\sigma_0} \cdot (1 + 2\sigma_0) = (1 + 2\sqrt{\rho^*})(1 + \sqrt{\rho^*}) = 1 + 3\sqrt{\rho^*} + 2\rho^*. \quad (3)$$

Now, by part (i) of Definition 2.1, e belongs to at most two of the trees in Γ. Thus by part (ii) of Definition 2.2, the total amount of flow across e due to the R-parts of all paths

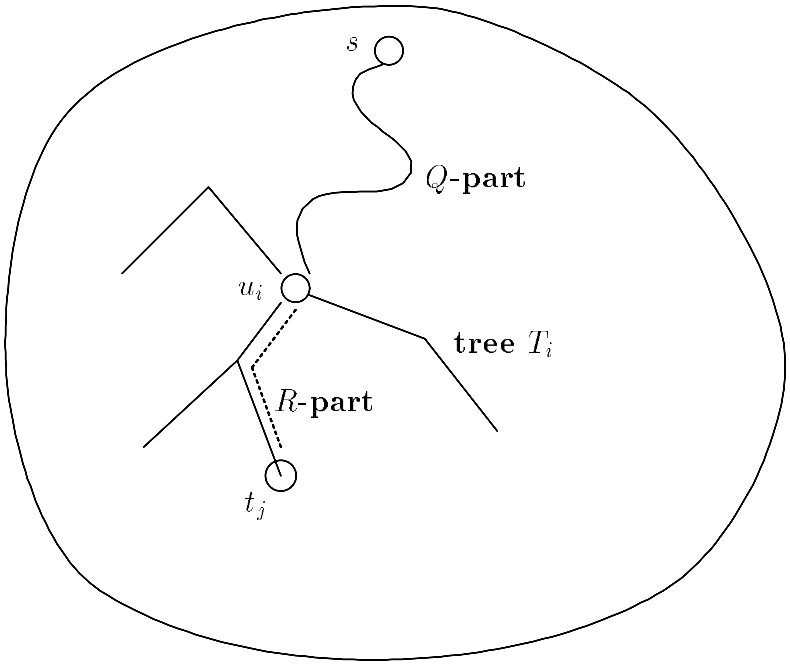

Figure 1: Constructing the unsplittable flow

is at most $2\sigma_0 + 2\rho^*$. Since $c_e \geq 1$, the total relative congestion on e due to the R-parts of all flow paths is at most

$$2\sigma_0 + 2\rho^* = \sqrt{\rho^*} + 2\rho^* \tag{4}$$

Adding (3) and (4), the total relative congestion on e is at most

$$1 + 4\sqrt{\rho^*} + 4\rho^*,$$

from which the theorem follows. ■

When $\rho^* = 1$, so that we are making no assumptions about having "small" demands, the congestion bound of Theorem 3.1 is equal to 9. We note that with a less clean proof, one can slightly improve the constant in Theorem 3.1, as follows. Our construction of strong coalitions in Lemma 2.3 results in at most one exceptional tree. As a consequence, if e is contained in two trees, then one is non–exceptional, and hence the node u_i for that tree will be a center node. From this it follows that the total flow across edge e due to the R-parts of all flow paths is at most $\sigma_0 + \rho^*$ (from the one exceptional tree that might contain e) plus $\frac{1}{2}\sigma_0 + \frac{1}{2}\rho^*$ (from the other tree containing e, which must be non–exceptional), and hence is at most $\frac{3}{2}\sigma_0 + \frac{3}{2}\rho^*$. Setting $\sigma_0 = \sqrt{\frac{2\rho^*}{7}}$, we obtain a bound of $1 + \sqrt{14\rho^*} + \frac{7}{2}\rho^*$, which is approximately 8.25 when $\rho^* = 1$. For the sake of simplicity, we will continue to use the bound stated in Theorem 3.1.

Of course, a natural goal is to determine the tightest possible bound for the minimum relative congestion of an unsplittable routing of $(G, \mathcal{D}, s)$, when it is fractionally feasible.

The following example shows that it must be at least $2 - o(1)$. Let G consist of vertices s and t, with k parallel edges of unit capacity between them. Let $\mathcal{D}$ consist of $k+1$ copies of the vertex t, with each terminal in $\mathcal{D}$ having demand $1 - \frac{1}{k+1}$. Then $(G, \mathcal{D}, s)$ is fractionally feasible, but the minimum congestion of an unsplittable routing of $(G, \mathcal{D}, s)$ is $2 - \frac{2}{k+1}$. Whether this example gives the worst possible gap between the fractional and unsplittable congestions is an interesting open question.

We say that the cut condition is *met by a factor of* β if $c(X) \geq \beta\rho(X)$ for all sets X not containing s. We now use Theorem 3.1 to show that if the cut condition is met by a large enough factor, then there is a feasible unsplittable flow.

Corollary 3.2 *Suppose that in $(G, \mathcal{D}, s)$ we have $\rho^* < \frac{1}{4}$, and the cut condition is met by a factor of at least $\beta = (1 - 2\sqrt{\rho^*})^{-2}$. Then $(G, \mathcal{D}, s)$ has a feasible unsplittable solution.*

Proof. We define G' to be the graph obtained from G' by dividing down all capacities by a factor of β. In the resulting problem $(G', \mathcal{D}, s)$, the ratio of the maximum demand to the minimum capacity is at most $\beta\rho^*$, and so by Theorem 3.1 there is an unsplittable flow for $(G', \mathcal{D}, s)$ of relative congestion at most $(1 + 2\sqrt{\beta\rho^*})^2 = \beta$. This is a feasible unsplittable flow in $(G, \mathcal{D}, s)$. (Note that we required $\rho^* < \frac{1}{4}$ so that β would be well–defined.) ∎

Finally, we give an approximation algorithm for the minimum congestion of a single–source unsplittable flow on an arbitrary graph G.

Corollary 3.3 *There is a 9–approximation to $\nu(G, \mathcal{D}, s)$ in an arbitrary graph G.*

Proof. We can assume that all edges have capacity 1, and that the maximum demand is $\rho^* < 1$. In polynomial time, we first compute a fractional flow in $(G, \mathcal{D}, s)$ of minimum congestion ν^*. We know that ν^* is a lower bound on the optimal unsplittable congestion, and we know that ρ^* is also a lower bound on this optimum (since some edge must carry ρ^* units of flow). Thus, we set $c = \max(\nu^*, \rho^*)$, and assign all edges a capacity of c. Since there is a feasible fractional flow with this choice of capacity, we know by Theorem 3.1 that there is an unsplittable flow of congestion at most $(1 + 2\sqrt{\rho^* c^{-1}})^2 \leq 9$, from which the approximation ratio follows. ∎

4 Maximization

We now turn from the problem of minimizing congestion to that of maximizing the total amount of flow that can be sent subject to the capacity constraints. Recall that the optimum for this problem, in the single–source case, is denoted $\alpha(G, \mathcal{D}, s)$. We obtain a constant–factor approximation for $\alpha(G, \mathcal{D}, s)$; and since our performance guarantee will in fact be in terms of the fractional optimum, it will imply a constant–factor gap between the values of the fractional and unsplittable optima.

Again, we let $(G, \mathcal{D}, s)$ denote a single–source UFPwith minimum edge capacity 1. If all demands have roughly the same value, then it is not difficult to obtain a constant–factor approximation algorithm by simply assigning the same demand value to all terminals and applying Theorem 1.1. This is the content of Lemmas 4.1 and 4.2.

We use these lemmas to take care of demands that are "sufficiently large" — in particular, those larger than a constant ρ_0 that we will define below. For the set of all demands smaller than ρ_0, we require a technique involving strong coalitions (Theorem 4.3). Specifically, we first compute a maximum flow on the set of leaders of a strong coalition; we then construct an unsplittable flow on the terminals in $\mathcal{D}$ so that for each leader routed by the initial maximum flow, we are able to route a constant fraction of the demand in its tree. This will provide us with a constant–factor approximation ratio.

Recall that $\rho^* = \max_i \rho_i$. We define $\rho_* = \min_i \rho_i$.

Lemma 4.1 *There is a polynomial–time algorithm producing an unsplittable flow of value at least $\frac{1}{2}\rho_*/\rho^*$ times $\alpha^f(G, \mathcal{D}, s)$.*

Proof. Write $y = \rho_*/\rho^*$. Let $(G, \mathcal{D}, s)$ be an instance of the single–source UFP. Let G' denote the graph obtained from G by first rounding all capacities down to the nearest multiple of ρ^*, and then multiplying them by y. Thus, each capacity in G' is between $\frac{1}{2}y$ and y times what it is in G. Let $\mathcal{D}'$ denote the set of terminals obtained from $\mathcal{D}$ by assigning each a demand of ρ_*. We first observe that $\alpha^f(G', \mathcal{D}', s)$ is at least $\frac{1}{2}y$ times $\alpha^f(G, \mathcal{D}, s)$. To see this, take any fractional flow in $(G, \mathcal{D}, s)$ and multiply all the flow values by a factor of $\frac{1}{2}y$. The resulting flow satisfies all capacity constraints in G', and it does not route more than ρ_* units of flow to any terminal in $\mathcal{D}'$; thus it is a feasible fractional flow in $(G', \mathcal{D}', s)$.

But in $(G', \mathcal{D}', s)$, all demands are equal to ρ_*, and all capacities are multiples of ρ_*; thus by Theorem 1.1, we can find an unsplittable flow of value equal to $\alpha^f(G', \mathcal{D}', s)$. The flow paths from this unsplittable solution are in turn feasible in $(G, \mathcal{D}, s)$, since the demands in $(G, \mathcal{D}, s)$ are at most a factor of y^{-1} greater, while the capacities are at least y^{-1} times greater. Thus, the value of this unsplittable flow is within a factor of $\frac{1}{2}y$ of $\alpha^f(G, \mathcal{D}, s)$. ■

Lemma 4.2 *There is a polynomial–time algorithm producing an unsplittable flow of value at least $[2e\lceil\ln(\rho^*/\rho_*)\rceil]^{-1}$ times $\alpha^f(G, \mathcal{D}, s)$.*

Proof. Write $q = \lceil\ln(\rho^*/\rho_*)\rceil$. We partition the terminals into q subsets $\mathcal{D}_1, \ldots, \mathcal{D}_q$, so that the demands of all terminals in $\mathcal{D}_i$ are in the interval $[\rho_* e^{i-1}, \rho_* e^i]$. In each of the problems $(G, \mathcal{D}_i, s)$, we can produce an unsplittable flow of value at least $(2e)^{-1}$ times $\alpha^f(G, \mathcal{D}_i, s)$, using Lemma 4.1. Since $\alpha^f(G, \mathcal{D}, s) \leq \sum_{i=1}^q \alpha^f(G, \mathcal{D}_i, s)$, we can obtain the claimed approximation ratio by taking, from among the q unsplittable flows that we compute, the one of the greatest value. ■

Theorem 4.3 *Let $\rho^* < \frac{3}{2} - \sqrt{2}$. There is a polynomial–time algorithm for the single–source* UFP *producing a flow of value at least $1 - 4\sqrt{\rho^*} + 2\rho^*$ times $\alpha^f(G, \mathcal{D}, s)$.*

Proof. We require $\rho^* < \frac{3}{2} - \sqrt{2}$ so that the expression $1 - 4\sqrt{\rho^*} + 2\rho^*$ is positive. Let $(G, \mathcal{D}, s)$ be an instance of the single–source UFP in which $\rho^* < \frac{3}{2} - \sqrt{2}$. We set $\sigma_0 = \frac{\sqrt{\rho^*}}{2 - 2\sqrt{\rho^*}}$; note that

$$\sigma_0 - \rho^* = \frac{\sqrt{\rho^*} - 2\rho^* + 2\rho^{*3/2}}{2 - 2\sqrt{\rho^*}} = \frac{(\rho^{*1/4} - \rho^{*3/4})^2 + \rho^{*3/2}}{2 - 2\sqrt{\rho^*}} > 0.$$

Let Γ be a strong coalition in $(G, \mathcal{D}, s)$ of order $\sigma_0 - \rho^*$, let $T_1, \ldots, T_r$ denote the trees in Γ, and for $i = 1, \ldots, r$ let $\mathcal{D}_i$ denote the set of terminals assigned to the tree T_i.

Let G' denote the graph obtained by multiplying all capacities of G by $1 + \sigma_0$ and then rounding up to the nearest multiple of σ_0. We assign each non–exceptional vertex in $U(\Gamma)$ a demand of σ_0, and each exceptional vertex in $U(\Gamma)$ a demand equal to the amount assigned to its tree. (Because of this latter point, we are in fact placing demand on s itself; this is simply to be able to make a comparison to the optimum, below.) We then compute a maximum unsplittable flow in $(G', U(\Gamma), s)$. Since the only demands not equal to σ_0 are situated at the source s, Theorem 1.1 implies that this maximum unsplittable flow is equal to the fractional optimum in $(G', U(\Gamma), s)$ and can be computed in polynomial time.

We claim that the value of this unsplittable flow is at least as large as the value of the maximum fractional flow in $(G, \mathcal{D}, s)$. For consider any fractional flow in $(G, \mathcal{D}, s)$, and let ε_i denote the demand contained in tree T_i of Γ that was routed by this flow. Then by Lemma 2.4, there is a feasible fractional flow in $(G', U(\Gamma), s)$ that routes ε_i demand from $u_i \in U(\Gamma)$. But the unsplittable flow that we compute in $(G', U(\Gamma), s)$ has value equal to that of the *maximum* fractional flow, proving the claim.

Let U' denote the set of non–exceptional vertices of $U(\Gamma)$ that are routed by the maximum unsplittable flow constructed in the previous paragraph, and I' the set of indices of such vertices in $\{1, \ldots, r\}$. For $i \in I'$, we define $\mathcal{D}'_i \subset \mathcal{D}_i$ to be any set which is maximal subject to the condition that

$$\rho(\mathcal{D}'_i) \leq \left(\frac{1 - 2\sigma_0}{1 + 2\sigma_0}\right) \sigma_0. \tag{5}$$

Note that by the maximality of $\mathcal{D}'_i$, we have

$$\rho(\mathcal{D}'_i) \geq \left(\frac{1 - 2\sigma_0}{1 + 2\sigma_0}\right) \sigma_0 - \rho^*. \tag{6}$$

Let $\mathcal{D}' = \cup_{i \in I'} \mathcal{D}'_i$. Finally, let I'' denote the set of indices of the exceptional trees, and $\mathcal{D}'' = \cup_{i \in I''} \mathcal{D}_i$.

We will show that $\mathcal{D}' \cup \mathcal{D}''$ is realizable in the graph G. We argued above that $\sigma_0 \cdot |I'| + \rho(\mathcal{D}'')$ is at least as large as the maximum fractional flow in $(G, \mathcal{D}, s)$; by Equation (6), the unsplittable flow that we produce has value at least

$$\left[\left(\frac{1 - 2\sigma_0}{1 + 2\sigma_0}\right) \sigma_0 - \rho^*\right] |I'| + \rho(\mathcal{D}'').$$

This value is thus within a factor of at least

$$\frac{\left(\frac{1-2\sigma_0}{1+2\sigma_0}\right) \sigma_0 - \rho^*}{\sigma_0} = 1 - \frac{\frac{2\sqrt{\rho^*}}{1-\sqrt{\rho^*}}}{\frac{1}{1-\sqrt{\rho^*}}} - \frac{2\rho^*}{\frac{\sqrt{\rho^*}}{1-\sqrt{\rho^*}}} = 1 - 4\sqrt{\rho^*} + 2\rho^*. \tag{7}$$

times that of the fractional optimum.

Thus it remains only to show that $\mathcal{D}' \cup \mathcal{D}''$ is realizable in G. For $u_i \in U'$, let Q_i denote the u_i-s flow path in the maximum unsplittable flow we constructed in $(G', U(\Gamma), s)$. Now, consider any $t_j \in \mathcal{D}' \cup \mathcal{D}''$, with $\tau(t_j) = T_i$. As in the proof of Theorem 3.1, we define R_j to be the unique t_j-u_i path in T_i. We then define P_j to be the concatenation of Q_i and R_j. P_j will serve as the flow path for the terminal t_j.

We claim that this set $\{P_j\}$ of flow paths does not violate any capacity constraint. To show this, we consider any edge e. We know that if each flow path Q_i were to carry σ_0 units of flow, then at most $(1+2\sigma_0)c_e$ units of flow would pass across e due to the Q-parts of all paths in $\{P_j\}$. But now, each flow path Q_i carries at most

$$\left(\frac{1-2\sigma_0}{1+2\sigma_0}\right)\sigma_0$$

units of flow, and hence at most $(1-2\sigma_0)c_e$ units of flow pass across edge e due to the Q-parts of all paths in $\{P_j\}$. Also, since e is contained in at most two trees of Γ, and $c_e \geq 1$, at most $2\sigma_0 \leq 2\sigma_0 c_e$ units of flow pass across e due to the R-parts of all paths in $\{P_j\}$. Hence, the capacity constraint for edge e is satisfed.

Thus the set $\mathcal{D}' \cup \mathcal{D}''$ is realizable in G, using the flow paths $\{P_j\}$. The theorem follows. ■

From this result, we can obtain a constant–factor approximation without any assumptions about the size of the largest demand (aside from $\rho_i \leq c_e$ for all i, e). Let $\mathcal{D}_0$ denote the set of terminals of demand less than e^{-3}, and $\mathcal{D}_1$ the set of terminals of demand at least e^{-3}. We can obtain a constant–factor approximation algorithm for $\alpha(G, \mathcal{D}_1, s)$ using Lemma 4.2, and a constant–factor approximation algorithm for $\alpha(G, \mathcal{D}_0, s)$ using Theorem 4.3. Of the two unsplittable flows produced this way, taking the one of greater value gives us

Corollary 4.4 *There is a constant–factor approximation algorithm for $\alpha(G, \mathcal{D}, s)$.*

5 Routing in Rounds

In order to obtain a constant–factor approximation for the problem of routing in rounds, we make use of a classical connection between flows and matroids [20, 28]. In the following section we discuss this connection and its consequences; we then present the approximation algorithm.

5.1 Tools from Matroid Theory

A *matroid* M is a pair $(U, \mathcal{M})$, where U is a finite ground set, and $\mathcal{M}$ is a collection of subsets of U that satisfies that following two conditions.

(i) *(Closure with respect to inclusion.)* If $Y \in \mathcal{M}$ and $X \subset Y$, then $X \in \mathcal{M}$.

(ii) *(The exchange property.)* If $X, Y \in \mathcal{M}$ and $|X| < |Y|$, then there exists an element $y \in Y$ such that $X \cup \{y\} \in \mathcal{M}$.

A subset of U is called *independent* if it belongs to $\mathcal{M}$; it is called a *basis* if it is a maximal independent set, with respect to inclusion. Note that the exchange property ensures that all bases have the same cardinality. If $X \subset U$, we define the *rank* of X, denoted $r(X)$, to be the maximum cardinality of an independent subset of X. We refer the reader to a book such as [28] for further background on matroids.

There are a number of basic problems that are computationally intractable for set systems in general, but which are tractable when the set system is a matroid. We will be primarily interested in the following *covering problem*: for a matroid $M = (U, \mathcal{M})$, what is the minimum cardinality cover of U by members of $\mathcal{M}$? This question is addressed by the following result of Edmonds [6].

Theorem 5.1 (Edmonds) *There is a polynomial–time algorithm to find a minimum cardinality cover of the ground set of a matroid $M = (U, \mathcal{M})$ by independent sets. Moreover, if this minimum is greater than j, then there is some $X \subset U$ for which $j \cdot r(X) < |X|$.*

We say that a single–source UFP is *integral* if all demands are equal to some common value ρ, and all capacities are multiples of ρ. Recall that by Theorem 1.1, such integral unsplittable problems are well–characterized by the max–flow min–cut theorem. The relevance of matroids to such problems is a consequence of the following result, due to Perfect [20].

Theorem 5.2 (Perfect) *Let $(G, \mathcal{D}, s)$ be an integral single–source UFP. The collection of realizable subsets of $\mathcal{D}$ forms a matroid over the ground set $\mathcal{D}$.*

The following fact is immediate.

Lemma 5.3 $\nu_0^f(G, \mathcal{D}, s) \leq \nu_0(G, \mathcal{D}, s) \leq \chi(G, \mathcal{D}, s)$.

We now show that when $(G, \mathcal{D}, s)$ is integral, the matroid structure of the problem implies that the minimum fractional relative congestion, rounded up to the nearest integer, is in fact equal to the minimum number of rounds required. This is far from the case for general unsplittable flow problems.

Lemma 5.4 *If $(G, \mathcal{D}, s)$ is integral, then $\lceil \nu_0^f(G, \mathcal{D}, s) \rceil = \chi(G, \mathcal{D}, s)$ (and hence the latter quantity can be computed in polynomial time).*

Proof. In view of the previous lemma, it is enough to show that $\chi(G, \mathcal{D}, s) \leq \lceil \nu_0^f(G, \mathcal{D}, s) \rceil$. Let us denote the former quantity by j. If $j = 1$, then we are done, so let us assume $j > 1$. By Theorem 5.2, we can view j as the minimum cardinality of a cover of $\mathcal{D}$ by members of the matroid of realizable subsets. Thus by Theorem 5.1, there is a set X such that $\rho(X)$ is more than $j - 1$ times as large as the maximum total demand of a realizable set of terminals in X. Thus, by Theorem 1.1, we have $c(X) < \frac{1}{j-1}\rho(X)$. It follows that $\nu_0^f(G, \mathcal{D}, s) > j - 1$, and so $\lceil \nu_0^f(G, \mathcal{D}, s) \rceil \geq j$. ■

5.2 The Approximation Algorithm

We are given a single–source problem $(G, \mathcal{D}, s)$; we assume by re-scaling that the minimum edge capacity is 1, and all demands are between 0 and 1. Our approximation algorithm will be similar to that of the previous section; we choose a threshold value, and treat demands larger than the threshold separately from those smaller than the threshold. As above, it is straightforward to handle the large demands (Lemmas 5.5 and 5.6).

Lemma 5.5 *There is a polynomial–time algorithm that routes $(G, \mathcal{D}, s)$ in at most $\lceil 2\frac{\rho^*}{\rho_*}\nu_0^f(G, \mathcal{D}, s) \rceil$ rounds.*

Proof. Let G' denote the graph obtained from G by rounding all capacities down to the nearest multiple of ρ^*, and $\mathcal{D}'$ denote the set of terminals obtained by assigning each terminal in $\mathcal{D}$ a demand of ρ^*. The problem $(G', \mathcal{D}', s)$ is integral; we compute a routing for this problem in $\chi(G', \mathcal{D}', s) = \lceil \nu_0^f(G', \mathcal{D}', s) \rceil$ rounds, and use the associated paths for the problem $(G, \mathcal{D}, s)$. Clearly the paths used in each round respect the capacity constraints in $(G, \mathcal{D}, s)$.

By an argument strictly analogous to the proof of Lemma 4.1, we have $\nu_0^f(G', \mathcal{D}', s) \leq 2\frac{\rho^*}{\rho_*}\nu_0^f(G, \mathcal{D}, s)$, from which the result follows. ■

Similarly, by an argument strictly analogous to the proof of Lemma 4.2, we have

Lemma 5.6 *There is a polynomial–time algorithm that routes $(G, \mathcal{D}, s)$ in at most*

$$\lceil \ln(\rho^*/\rho_*) \rceil \cdot \lceil 2e\nu_0^f(G, \mathcal{D}, s) \rceil$$

rounds.

Theorem 5.7 *Let $\rho^* < \frac{7}{4} - \sqrt{3}$. There is a polynomial–time algorithm that routes $(G, \mathcal{D}, s)$ in $2\lceil \nu_0^f(G, \mathcal{D}, s) \rceil$ rounds.*

Proof. Let $(G, \mathcal{D}, s)$ be an instance of the single–source UFP in which $\rho^* < \frac{7}{4} - \sqrt{3}$. We set $\sigma_0 = \frac{\sqrt{\rho^*}}{2 - 2\sqrt{\rho^*}}$ and let Γ be a strong coalition in $(G, \mathcal{D}, s)$ of order $\sigma_0 - \rho^*$. Let $T_1, \ldots, T_r$ denote the trees in Γ, and for $i = 1, \ldots, r$ let $\mathcal{D}_i$ denote the set of terminals assigned to the tree T_i.

Let G' denote the graph obtained by multiplying all capacities of G by $1 + \sigma_0$ and then rounding up to the nearest multiple of σ_0. We assign each non–exceptional vertex in $U(\Gamma)$ a demand of σ_0, and each exceptional vertex in $U(\Gamma)$ a demand of 0. We claim that there is a fractional routing of $(G', U(\Gamma), s)$ with relative congestion at most $\lceil \nu_0^f(G, \mathcal{D}, s) \rceil$. To see this, note that if we were to multiply all capacities of G by a factor of $\lceil \nu_0^f(G, \mathcal{D}, s) \rceil$, then there would be a feasible fractional routing of $\mathcal{D}$, *and* all capacities would be at least 1; thus, by Lemma 2.4, there would be a fractional routing of $U(\Gamma)$ of relative congestion at most $1 + \sigma_0$. But the capacities in G' are at least $1 + \sigma_0$ times what they are in G, and hence if we were to multiply all capacities of G' by a factor of $\lceil \nu_0^f(G, \mathcal{D}, s) \rceil$, then $U(\Gamma)$ would be fractionally feasible. It follows that $\nu_0^f(G', U(\Gamma), s) \leq \lceil \nu_0^f(G, \mathcal{D}, s) \rceil$.

The algorithm is now as follows. We compute a routing of $(G', U(\Gamma), s)$ in $\lceil \nu_0^f(G', U(\Gamma), s) \rceil \leq \lceil \nu_0^f(G, \mathcal{D}, s) \rceil$ rounds. Now, we break each of these rounds into two rounds: if u_i is routed in a given round j, then we route the terminals in its tree in the pair of rounds associated with j. This will result in the claimed approximation ratio. It therefore remains to show how to take a single round of our routing of $(G', U(\Gamma), s)$, and route the associated terminals in two rounds in $(G, \mathcal{D}, s)$.

Thus, let U' denote the set of vertices in $U(\Gamma)$ routed in a single round of the routing for $(G', U(\Gamma), s)$, and I' the set of indices of such vertices. Let $\mathcal{D}_i$ denote the of terminals associated with a vertex $u_i \in U$. Since the total demand of terminals in $\mathcal{D}_i$ is at most σ_0, and the maximum demand is ρ^*, we can partition $\mathcal{D}_i$ into two sets, $\mathcal{D}_i^1$ and $\mathcal{D}_i^2$ so that the demand in each is at most $\frac{1}{2}\sigma_0 + \rho^*$. Recall from Equation (7) that

$$\frac{\left(\frac{1-2\sigma_0}{1+2\sigma_0}\right)\sigma_0 - \rho^*}{\sigma_0} = 1 - 4\sqrt{\rho^*} + 2\rho^*. \tag{8}$$

Now since $\rho^* < \frac{7}{4} - \sqrt{3}$, we have $1 - 4\sqrt{\rho^*} + 2\rho^* < \frac{1}{2}$, and hence by Equation (8) we have

$$\rho(\mathcal{D}_i^1), \rho(\mathcal{D}_i^2) \leq \frac{1}{2}\sigma_0 + \rho^* \leq \left(\frac{1 - 2\sigma_0}{1 + 2\sigma_0}\right)\sigma_0.$$

We claim that for $\ell = 1, 2$, the set of terminals in $\mathcal{D}_i^\ell$ $(i \in I')$ can be routed in a single round. To show this, we define the following flow path P_j for $t_j \in \mathcal{D}_i^\ell$: we concatenate the t_j-u_i path in T_i, denoted R_j, with the u_i-s flow path, denoted Q_i. Now for any edge e, the total amount of flow crossing e due to Q-parts of paths is at most

$$(1 + 2\sigma_0)c_e \cdot \left(\frac{1 - 2\sigma_0}{1 + 2\sigma_0}\right) = (1 - 2\sigma_0)c_e,$$

and the total amount of flow crossing e due to R-parts of paths is at most $2\sigma_0 \leq 2\sigma_0 c_e$. Thus the capacity constraints are satisfied for the set of paths $\{P_j\}$ associated with terminals in $\cup_{i \in I'} \mathcal{D}_i^\ell$.

Finally, since terminals in exceptional trees are routed entirely on paths of the form R_j, they can all be routed in one of the rounds already used, without violating the capacity constraints. ■

The above facts provide us with the main approximation result of this section — namely, an approximation algorithm for $\chi(G, \mathcal{D}, s)$ when ρ_* and ρ^* are arbitrary numbers in $(0, 1]$. The approximation is obtained simply by using separate sets of rounds for the terminals of demand at least $\frac{7}{4} - \sqrt{3}$ and for those of demand less than $\frac{7}{4} - \sqrt{3}$. As we are within a constant factor of optimal for each type of demand, we have

Corollary 5.8 *Let $(G, \mathcal{D}, s)$ be an arbitrary single–source unsplittable flow problem. There is a polynomial–time constant–factor approximation for $\chi(G, \mathcal{D}, s)$.*

It is also worth noting another natural corollary of the above results.

Corollary 5.9 *If $(G, \mathcal{D}, s)$ is fractionally feasible, then it is routable without splitting in a constant number of rounds. If additionally $\rho^* < \frac{7}{4} - \sqrt{3}$, then it is routable without splitting in two rounds.*

6 Directed Graphs

We now extend the approximation algorithms of the previous sections to the case of directed graphs. Thus, let $G = (V, E)$ be a directed graph; s a distinguished vertex of G; and $\mathcal{D} = \{t_1, \ldots, t_k\}$ a set of terminals in G with associated demands $\rho_1, \ldots, \rho_k$. Unless we say otherwise, *paths* and *cycles* in G will be directed. We seek directed paths *from s to* the terminals in $\mathcal{D}$ such that the capacity constraints are met. We again are able to obtain constant approximation ratios that converge to 1 as the ratio of the maximum demand to the minimum capacity goes to 0. The underlying technique here is still based on the construction of coalitions, now using directed *arborescences* in place of trees. However, one must be more careful in defining the "right" coalition in the directed case; we will discuss this below.

There are two further points worth noting here about the relation of the directed case of this problem to the undirected case. First, the disjoint paths problem is much less well understood for directed graphs than it is for undirected graphs, and so it is of particular interest to develop approximation techniques that are applicable to directed graphs. In keeping with this general principle, we will find that the algorithms we obtain for directed graphs will be more complicated than the ones just presented for the undirected case.

Second, we noted in the introduction that one can reduce a number of NP-hard load balancing problems to the basic single–source unsplittable flow problem; but most of the natural reductions here require that the underlying graph be directed.

As a basic example, consider the following machine scheduling problem. We have a set $J = \{j_1, \ldots, j_p\}$ of *jobs*, such that job j_i has *weight* w_i, and a set $M = \{m_1, \ldots, m_q\}$ of *machines*. Associated with job j_i is a set $S_i \subset M$ of machines on which it can run. The goal is to assign each job to a machine in such a way that the maximum weight assigned to any machine is minimized. In the case in which all S_i are equal to M, this is just the classical *minimum makespan problem*, for which a $(1 + \varepsilon)$–approximation can be found in polynomial time for any fixed $\varepsilon > 0$ [12]. But when the S_i are arbitrary sets, Lenstra, Shmoys, and Tardos show that no polynomial–time algorithm can approximate the optimum load to within a factor better than $\frac{4}{3}$ [18]; and they provide a 2-approximation algorithm for a class of problems that includes this one.

Here, we observe that this load balancing problem can be easily formulated as a single–source unsplittable flow problem on a three–level directed graph, as in Figure 2. We introduce a node for every $m_j \in M$ and $j_i \in J$, and we include a directed edge (m_j, j_i) iff $m_j \in S_i$. We then attach a source s by an edge to each $m_j \in M$, and give each j_i a demand equal to w_i. Then the minimum–congestion routing of an unsplittable flow in this graph corresponds to a minimum–load assignment of jobs to machines. (We note that the constant factor of approximation we obtain for the general single–source problem is greater than 2, so we are not claiming to improve on the bound of Lenstra, Shmoys, and Tardos for this special case.)

Of course, much more elaborate reductions are possible. As another example, one could consider a *multi-stage load balancing* problem in which there is a set J of jobs, and sets $S_1, \ldots, S_k$ of *sites*, residing in a metric space M. Each job must be processed by one site from each set S_i, in sequence; but in moving from one site to the next, a job cannot move more than some distance Δ in the metric space M. The goal is to find a way of processing each job, so that the maximum load on any site is minimized. Again, it is straightforward to formulate this as the problem of finding the minimum congestion of a single–source unsplittable flow in an appropriate directed graph.

Definitions and Techniques

For directed graphs, there is a version of the max-flow min-cut theorem strictly analogous to Theorem 1.1, in terms of the following definitions. Let $G = (V, E)$ be a directed graph, with source vertex s, and $\mathcal{D}$ a set of terminals in G. If $X \subset V$, we define $\delta^-(X)$ to be the set of edges entering X, and $\delta^+(X)$ to be the set of edges leaving X; we define $c^-(X)$ and $c^+(X)$ to be the total capacity of edges entering and leaving X, respectively. Finally, we define $\rho(X)$ to be the sum of all demands of terminals in $\mathcal{D} \cap X$. A natural phrasing of the cut condition

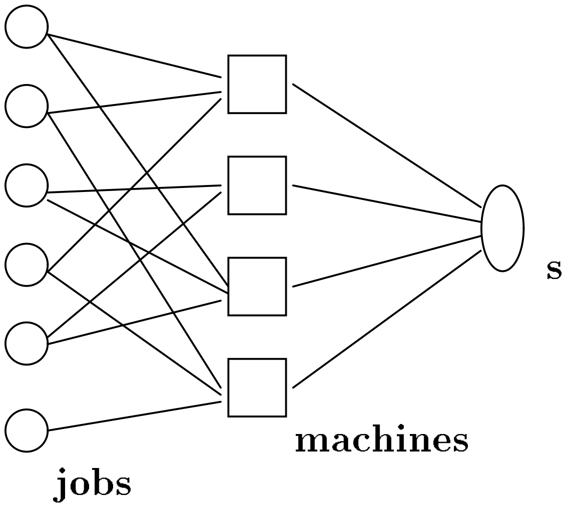

Figure 2: A load balancing problem

is as follows: for all $X \subset V \setminus \{s\}$, we have $c^{-}(X) \geq \rho(X)$. The max-flow min-cut theorem for directed graphs can now be stated as follows.

Theorem 6.1 (Ford–Fulkerson) *Let G be an arbitrary directed graph, s a source vertex, and $\mathcal{D}$ a set of terminals with demands.*

(i) There is a fractional routing from s to $\mathcal{D}$ if and only if the cut condition holds.

(ii) A maximum fractional flow can be found in polynomial time.

(iii) If all demands ρ_i are equal to a common value ρ, and all capacities are multiples of ρ, then there is a maximum fractional flow that is an unsplittable flow. Moreover, this unsplittable flow can be found in polynomial time.

In this section, we will work with graphs whose capacity values are integers; by allowing parallel edges, we can assume that every capacity is equal to 1.

As in the undirected case of the single-source unsplittable flow problem, we make use of the notion of *coalitions* as a way of "grouping" demand in G. However, since G is now a directed graph, we must make sure that the leader of each tree in the coalition has a directed path to each of its terminals — in this way, the demand that has been grouped into this leader node can still reach the terminals in which it resides. Thus, we first define the types of directed trees that we will be dealing with, and then discuss some of the additional complications that arise when using directed coalitions.

Recall that we are interested in finding flow paths from s to the terminals $\mathcal{D} = \{t_1, \ldots, t_k\}$. Thus we will assume that there is a path from s to every other node of G; this is no loss of generality, since no other nodes can participate in a solution to the unsplittable flow problem defined by $(G, \mathcal{D}, s)$. By a *tree* in G, we mean a subgraph T of G whose underlying edge set forms a tree if one ignores the directions of the edges. If T is a tree in G, and v is a vertex in T with a path to every other vertex in T, then we say that v is a *root* of T. A tree can have at most one root; if a tree T has a root, then we will say that T is an *arborescence*. We say that T is a *spanning arborescence* if it is an arborescence whose vertex set is equal to V. Finally, a *forest* is a set of trees in G that do not share vertices.

Given these definitions, a directed coalition will simply be a set of half–disjoint arborescences, with each terminal assigned to an arborescence that contains it. Strong directed coalitions can be constructed in a straightforward fashion from a depth-first search of any spanning arborescence of G rooted at s (Lemma 6.4).

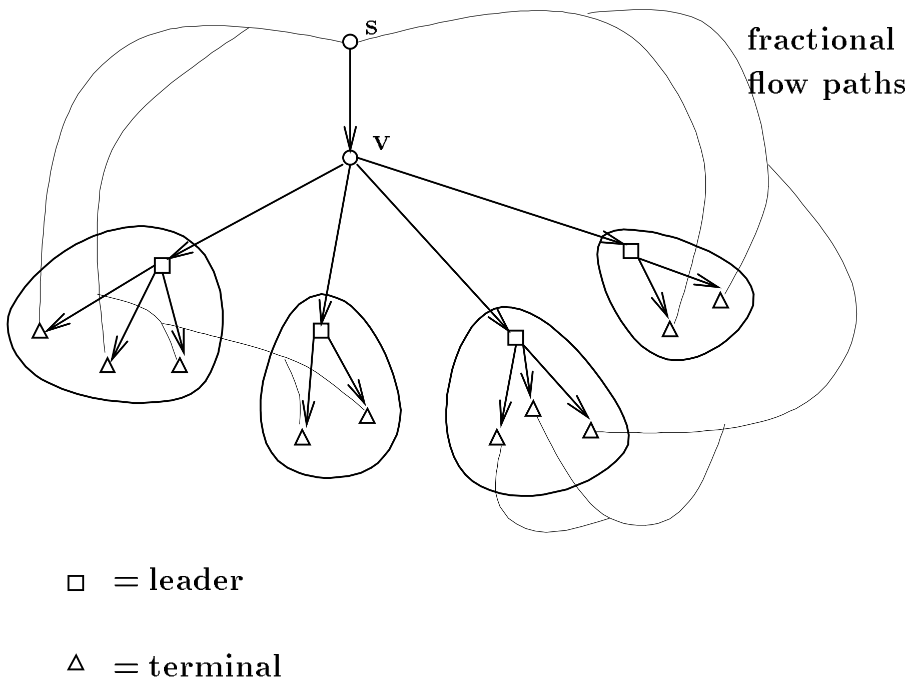

Figure 3: A bad choice of coalition

The main technical complication arises from the fact that the obvious analogue of Lemma 2.4 does not hold in the directed case — given a fractionally realizable set of terminals, it is possible to construct a coalition so that the resulting set of leaders can only be routed with a very large congestion. Such an example is pictured in Figure 3: although there are flow paths to all the terminals, the coalition is built out of edges not participating in this flow, and all the paths from s to the leaders share a single edge (s, v); this results in very high congestion. Note that this difficulty cannot arise in the undirected case, where a set of low–congestion paths to the leaders could be built by taking the fractional flow paths to the terminals in the reverse direction; this is precisely what is taking place in the proof of Lemma 2.4.

What we will show, in Lemma 6.8, is that one can always choose a "good" coalition, so that the leaders can be routed with approximately the same congestion as the terminal set. Given this lemma, the approximation algorithms will follow almost exactly as in the undirected case.

The construction of a good coalition proceeds roughly as follows. By the max-flow min-cut theorem for directed graphs (Theorem 6.1), we have to avoid the following situation:

there is a set $S \not\ni s$ containing a large number of leaders, such that very few edges enter S. Thus, given a set S containing a large number of leaders (but not s), we consider the following two cases:

(i) S also contains a correspondingly large number of terminals. In this case, since the terminal set is fractionally routable, there must be many edges entering S.

(ii) S contains very few terminals. Since there are paths from the leaders to their associated terminals, and very few of these terminals lie in S, it must be that there are many edges *leaving* S. In this case, we examine the set of edges leaving S. We say that an edge is *dense* if it carries at least $\frac{1}{2}$ units of flow from s to the terminal set, and *shallow* otherwise. We arrange the construction of our coalition $\vec{\Gamma}$ so that if a node has any entering edge that is dense, such an edge enters it in the arborescences of $\vec{\Gamma}$. We again have two cases to consider.

   (a) If a large fraction of the edges leaving S are dense, then a large amount of flow must be leaving S. It follows that a large amount of flow also enters S (since $s \notin S$), and hence there are many edges entering S.

   (b) Otherwise, suppose that most of the edges leaving S are shallow. This is the most difficult case, since we cannot argue that a lot of flow enters S. However, if we grow the set S by following chains of vertices of in–degree 1, and it remains the case that most of the edges leaving S are shallow, then by the construction of the coalition $\vec{\Gamma}$, we can argue that S now has out–edges to a large number of *shallow branching nodes* — nodes of in–degree greater than 1, all of whose entering edges are shallow.

   Now, the crucial fact that we establish in Lemma 6.8, using Lemmas 6.5 and 6.7, is that there are edge–disjoint paths linking s to *all* the shallow branching nodes in G. Consequently, we will be able to argue that when most of the edges leaving S are shallow, S must be entered by many of these edge–disjoint paths, and consequently there are many edges entering S, as required.

6.1 Coalitions

We first define coalitions in the directed setting.

Definition 6.2 *A* (weak) coalition *in* $(G, \mathcal{D}, s)$ *is a triple* $(\{A_1, \ldots, A_r\}, \{u_1, \ldots, u_r\}, \tau)$, *where each* A_i *is an arborescence in* G *with root* u_i, $\tau : \mathcal{D} \to \{A_1, \ldots, A_r\}$ *is a function, and the following properties are satisfed.*

(i) The arborescences are half–disjoint, *by which we mean that each edge of* G *belongs to at most two of the* A_i.

(ii) For each $i = 1, \ldots, k$, $t_i \in \tau(t_i)$.

Definition 6.3 *A* strong coalition *in* $(G, \mathcal{D}, s)$ *of order* σ *is a triple* $(\{A_1, \ldots, A_r\}, \{u_1, \ldots, u_r\}, \tau)$, *such that the following properties are satisfed.*

(i) $(\{A_1, \ldots, A_r\}, \tau)$ *is a coalition in* G.

(ii) For each $j = 1, \ldots, r$, either $u_j = s$ or we have

$$\sigma \leq \sum_{i:\tau(t_i)=A_j} \rho_i \leq \sigma + \rho^*. \tag{9}$$

As before, an arborescence for which inequality (9) does not hold will be called *exceptional.* For $i = 1, \ldots, r$, we say that u_i is the *leader* of A_i.

We show that strong coalitions exist in directed graphs. Our construction must be somewhat different from that of Section 2, since we need to produce arborescences rather than just trees. Thus, we proceed as follows. If A is an arborescence, and X is a set of vertices of A, consider the collection of subgraphs of A with property that they are arborescences containing X. It is easy to verify that this collection contains a unique minimal arborescence, which we will denote by $A[X]$.

Lemma 6.4 *Let A be an spanning arborescence in G with root s, and let $\sigma \geq 0$. Then there exists a strong coalition $\vec{\Gamma}$ in $(G, \mathcal{D}, s)$ of order σ, such that the edge sets of all arborescences in $\vec{\Gamma}$ are subsets of A.*

Proof. We perform a depth–first search of A, starting from the root s, and let $t_{\pi(1)}, \ldots, t_{\pi(k)}$ denote the order in which the terminal are first encountered in this depth–first search. As in the proof of Lemma 2.3, we partition this sequence into intervals $\mathcal{D}_1, \ldots, \mathcal{D}_r$, such that the total demand in each interval except possibly the last is between σ and $\sigma + \rho^*$. We add s to the set $\mathcal{D}_r$. We now construct the coalition $\vec{\Gamma}$ by defining, for $i = 1, \ldots, r$, $A_i = A[\mathcal{D}_i]$. We define u_i to be the root of the i^{th} arborescence, and the function τ so that $\tau(t_j) = A_i$ if $t_j \in \mathcal{D}_i$.

We show that the arborescences $\{A_i\}$ are half–disjoint; the other properties of Definitions 6.2 and 6.3 are straightforward to verify. Let C_i denote the set of edges traversed between the first visit to a vertex in $\mathcal{D}_i$ and the last visit to a vertex in $\mathcal{D}_i$. Now the sets $C_1, \ldots, C_r$ are half–disjoint, since every edge of A is traversed exactly twice in a depth–first search. But C_i contains the edge set of an arborescence on $\mathcal{D}_i$, and thus the edges of $A[\mathcal{D}_i]$ belong to C_i. It follows that the $\{A_i\}$ are half–disjoint. ■

Given any spanning arborescence A rooted at s, and any $\sigma \geq 0$, this lemma gives us a canonical strong coalition of order σ induced by A. Let us use $\vec{\Gamma}(A, \sigma)$ to denote this coalition. Now, at the beginning of this section, we indicated that the natural analogue of Lemma 2.4 does *not* hold in the directed case, and that we must use information about the fractional flow from s to the terminal set $\mathcal{D}$ in our construction of a directed coalition. In particular, we will need to prove the existence of disjoint paths from s to the *shallow branching nodes* of G, and for this we need Lemmas 6.5 and 6.7 below.

For a vertex v of a directed graph, let δ_v^+ denote the number of edges for which it is the tail, and δ_v^- denote the number of edges for which it is the head. Let $\delta_v = \delta_v^+ + \delta_v^-$. We say that v is a *leaf* if $\delta_v = 1$; if v is a leaf, we call it an *in-leaf* if $\delta_v^- = 1$, and an *out-leaf* if $\delta_v^+ = 1$. Finally, we say that T is a *reverse arborescence* with root u if reversing the direction of every edge turns it into an arborescence with root u.

Lemma 6.5 *Let H be a directed acyclic graph, and u a node of H, such that every node $v \in H \setminus \{u\}$ satisfies $\delta_v^- \geq \delta_v^+$. Then there exists a set of edge disjoint paths in H, such that every path has its tail in u, and for every $v \in H \setminus \{u\}$, at least $\delta_v^- - \delta_v^+$ paths have their heads in w.*

Proof. We add a new node u' to H, and for every $v \in H \setminus \{u\}$, we add $\delta_v^- - \delta_v^+$ parallel edges from v to u'. We also add δ_u^+ parallel edges from u' to u. Denote the resulting graph by H'. Note that in H', the in–degree of every node is equal to its out–degree, and so H' is Eulerian. Let C be an Eulerian circuit in this graph. If we delete u' from H', then C decomposes into edge–disjoint paths. It is easy to verify that these paths satisfy the conditions of the lemma. ■

Lemma 6.6 *Let A' be a reverse arborescence on at least two nodes, with root v, and let L be its set of out–leaves. Then there exists a collection of edge–disjoint paths in A' such that each path has its tail in a distinct vertex of L, at least δ_v^- paths have their heads in v, and for every $w \in A' \setminus (Y \cup \{v\})$, at least $\delta_w^- - 1$ of the paths have their heads in w.*

Proof. Attach a new node u to A', by a single edge to each node in L, and apply Lemma 6.5. ■

Lemma 6.7 *Let F be a forest, with vertex set W, and let $B = \{w \in W : \delta_w^- = 0\}$. Then there exists a collection of edge–disjoint paths in T such that*
(i) each path has its tail in a distinct vertex of B, and
(ii) for every $x \in W \setminus B$, at least $\delta_x^- - 1$ paths have their heads in x.

Proof. It is enough to prove this for a tree T; one can then apply the lemma in each component of F separately. Assuming T has more than one node, we fix an arbitrary node $u \notin B$. We root the tree at u; by this we mean that w will be defined to be *below* v if either (v, w) or (w, v) is an edge of T, and the undirected distance from u to v is less than the undirected distance from u to w. Say that an edge is *proper* if its tail is below its head, and *improper* otherwise. Let $T_1, \ldots, T_q$ be the set of trees obtained by deleting all improper edges.

We now observe the following facts.

(i) Each T_i is a reverse arborescence, since any node $v \in T_i$ can be the tail of at most one edge in T_i.

(ii) A node is a out–leaf of some T_i if and only if it belongs to the set B.

(iii) The in-degree of a node $v \neq u$ in T_i is equal to $\delta_v^- - 1$ if it is the root of T_i, and equal to δ_v^- otherwise.

(iv) The in–degree of u in its tree T_i is equal to δ_u^-.

From these four facts, it follows that applying Lemma 6.6 to each of the T_i produces the desired paths in T. ■

We now use these lemmas to construct an appropriate coalition for $(G, \mathcal{D}, s)$.

Lemma 6.8 *Let $(G, \mathcal{D}, s)$ be a single–source fractional flow problem, in which terminal $t_j \in \mathcal{D}_j$ has an associated demand ρ_j. Let us assign to each terminal in $\mathcal{D}$ a demand $\tilde{\rho}_j \leq \rho_j$, in such a way that $(G, \mathcal{D}, s)$ has a feasible fractional routing. Then there exists a spanning arborescence A of G, rooted at s, such that for every $\sigma \geq 0$, the induced coalition $\vec{\Gamma} = \vec{\Gamma}(A, \sigma)$ has the following property. Suppose we define the problem $(G, U(\vec{\Gamma}), s)$ by assigning to each $u_i \in U(\vec{\Gamma})$ any demand ρ'_i not exceeding the total $\tilde{\rho}$–demand of terminals in its arborescence. If for all i we have $\rho'_i \leq \varepsilon$, then $(G, U(\vec{\Gamma}), s)$ has a fractional routing of congestion at most $1 + 6\varepsilon$.*

Proof. By assigning a demand $\tilde{\rho}_j \leq \rho_j$ to each $t_j \in \mathcal{D}$, the resulting problem $(G, \mathcal{D}, s)$ has a fractional routing that respects the capacity constraints. Let f_e denote the total amount of flow crossing edge e in this fractional routing. We call an edge *tight* if $f_e = 1$, and *empty* if $f_e = 0$. We may delete any empty edges from G, as this does not affect the feasibility of $(G, \mathcal{D}, s)$ with demand $\tilde{\rho}$.

We first perform the following reduction steps on the graph G and the associated fractional flow.

(i) We can assume that the set of non–empty edges does not induce any directed cycles, and hence (deleting empty edges if necessary) that the graph G is acyclic and $\delta_s^- = 0$.

(ii) Suppose there exists a set of edges C that induce a simple cycle if one ignores their directions. Let $v_0, \ldots, v_{q-1}$ denote the set of nodes in this cycle, with each $e \in C$ having ends $v_i, v_{(i+1 \bmod q)}$ for some i. We say that the edge $(v_i, v_{(i+1 \bmod q)}) \in C$ is *forward* if its tail is v_i, and *backward* otherwise. We uniformly increase the flow values on forward edges in C while uniformly decreasing the flow values on backward edges, until some edge in C becomes tight or empty.

(iii) If there are two tight edges (u, v) and (v, w), then we may delete both from G and add the tight edge (u, w). If we find the desired coalition in the resulting graph, then it is easy to produce a coalition in the graph G.

Following these reduction steps, we may assume that G is acyclic, $\delta_s^- = 0$, the set of non-tight edges induce a forest F, and no node is both the head and the tail of a tight edge. Given the last point, let X denote the set of vertices, other than s, that are tails of tight edges. Let Y denote the set containing s and all vertices that are heads of tight edges. Note that X and Y are disjoint sets.

We say that an edge e of G is *dense* if $f_e \geq \frac{1}{2}$, and *shallow* otherwise. We say that a node v is *branching* if $\delta_v^- > 1$, and *shallow* if all edges for which it is the head are shallow. Let Z denote the set of all shallow, branching nodes.

For $u \in X$, let α_u denote the number of tight edges for which u is the tail. Thus at least α_u units of flow leave u; by flow conservation, at least α_u must enter u. Since this flow must enter on non–tight edges, the in–degree of u is at least $\alpha_u + 1$. In particular, u is branching. Moreover, if $u \in X$ is shallow (and hence in $Z \cap X$), then it must have in–degree at least $2\alpha_u + 1 \geq \alpha_u + 2$.

See Figure 4 for an example of all these definitions. To prove Lemma 6.8, we first show

(A) Z can be linked to s by edge–disjoint paths.

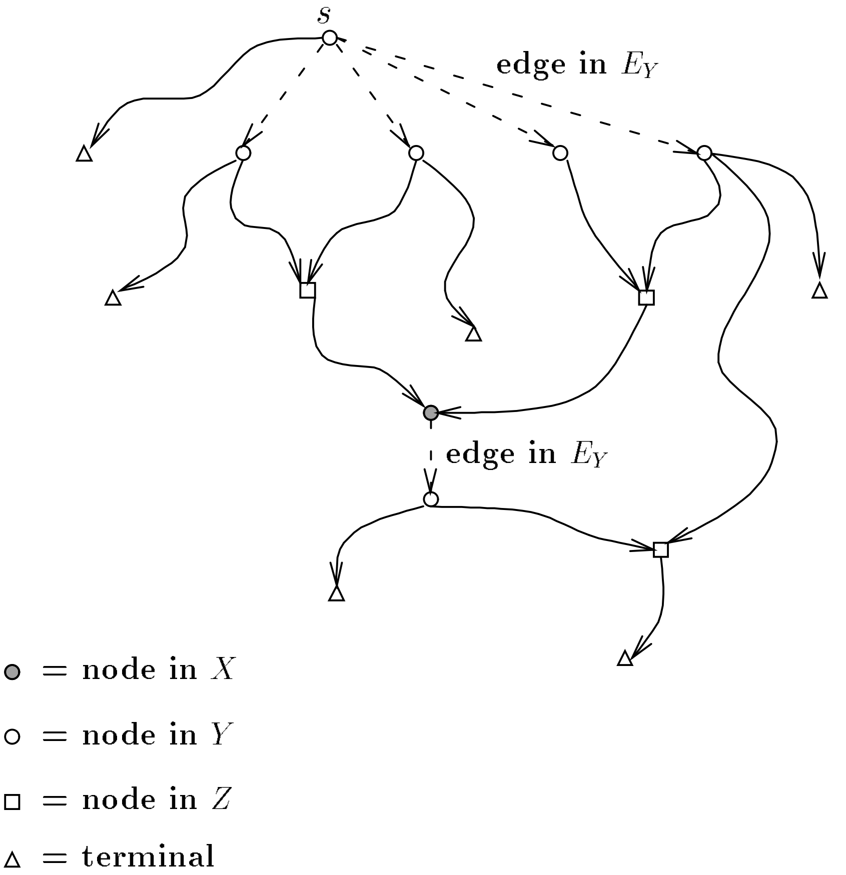

Figure 4: Constructing a good coalition

We prove (A) as follows. Let F denote the forest consisting of the non–tight edges and all nodes incident to them. We apply Lemma 6.7 to the forest F, obtaining a set of edge–disjoint paths $\mathcal{P}$. Any node v whose in–degree in F is zero must, by flow conservation, either be the head of a tight edge or be equal to s. Thus such a node belongs to Y, and so by Lemma 6.7, each of the paths in $\mathcal{P}$ has its tail in a distinct vertex of Y. Any node $u \in X \cup Z$ is not the head of any tight edge, and hence has the same in–degree in both G and F; thus u is the head of at least $\delta_u^- - 1$ paths in $\mathcal{P}$.

For each node $y \in Y \setminus \{s\}$, we now choose a single tight edge whose head is y, obtaining a collection of tight edges E_Y. Let E' denote the union of E_Y with the edges of $\mathcal{P}$, and let G' denote the subgraph of G consisting of the edges of E' and all nodes incident to them. For a node v of G', let $\delta^-_{v,G'}$ and $\delta^+_{v,G'}$ denote its in–degree and out–degree in the subgraph G'.

We claim that G', with source s, satisfies the hypotheses of Lemma 6.5. To see this, let v be any node of $G' \setminus \{s\}$; we show that $\delta^-_{v,G'} \geq \delta^+_{v,G'}$.

(i) If $v \in Y$, then it is at the tail of at most one path in $\mathcal{P}$, and at the head of one edge in E_Y.

(ii) If $v \in X$, then it is at the tail of at most α_v edges in E_Y, and at the head of at least $\delta_v^- - 1 \geq \alpha_v$ paths in $\mathcal{P}$.

(iii) If $v \notin X \cup Y$, then any edge for which it is the tail corresponds to an internal node of some path in $\mathcal{P}$, and can therefore be paired with some edge for which it is the head.

Let $\mathcal{Q}$ denote the set of edge–disjoint paths, anchored in s, that is provided by Lemma 6.5. We claim that for every $z \in Z$, the in–degree of z in G' is strictly greater than its out–degree; this will show that at least one path in $\mathcal{Q}$ has its head in z. To show this, we observe that if $z \in Z \setminus X$, then the edges of G' incident to it correspond to paths in $\mathcal{P}$ for which z is internal, and at least one path in $\mathcal{P}$ for which z is the head; thus $\delta^-_{z,G'} > \delta^+_{z,G'}$. And if $z \in Z \cap X$, then $\delta_z^+ \geq \alpha_z + 2$, and so z is the head of at least $\alpha_z + 1$ paths in $\mathcal{P}$. But z is the tail of at most α_z edges in E_Y, and so again we have $\delta^-_{z,G'} > \delta^+_{z,G'}$.

Thus there exists a set of edge–disjoint paths $\mathcal{Q}' \subseteq \mathcal{Q}$, with their tails in s, such that for every $z \in Z$, some path in $\mathcal{Q}'$ has its head in z. This proves Claim (A).

In this set of edge–disjoint s–Z paths, let P_z denote the path with ends s and $z \in Z$. We can now define the arborescence A. Since G is an acyclic graph in which s has a path to every other vertex, it is enough, in defining an arborescence, to specify an in–edge for every node of $V \setminus \{s\}$. For A, we do this as follows. Let $v \in V \setminus \{s\}$.

(i) If v is not a branching node, then we choose the unique edge whose head is equal to v.

(ii) If v is at the head of a dense edge, then we choose one of these dense edges arbitrarily.

(iii) Otherwise, v is a shallow branching node. In this case, we choose the final edge of the path P_v, constructed in the proof of Claim (A).

We now prove the main statement of the lemma. Let ρ'_i denote the demand assigned to $u_i \in U(\vec{\Gamma})$ in the problem $(G, U(\vec{\Gamma}), s)$. By hypothesis,

$$\rho'_i \leq \sum_{j:\tau(t_j)=A_i} \tilde{\rho}_j \leq \varepsilon. \tag{10}$$

We claim that if every edge is given a capacity of $1 + 6\varepsilon$, then the problem $(G, U(\vec{\Gamma}), s)$ is fractionally feasible. In order to show this, we must verify the cut condition, by showing that $(1 + 6\varepsilon)|\delta^-(S)| \geq \rho'(S)$ for every set $S \not\ni s$.

Suppose that there is a set violating this cut condition; let us choose such a set S whose cardinality is maximal. Suppose that S contains the leaders $u_{i_1}, \ldots, u_{i_q}$. For each of the associated arborescences A_{i_j}, we say that A_{i_j} is *unsplit* if its node set is entirely contained in S, and *split* otherwise. We consider the following two cases.

(i) At least a $\frac{1}{1+6\varepsilon}$ fraction of the total ρ'-demand of S is associated with leaders of unsplit arborescences. Thus, by Equation (10), we have

$$\tilde{\rho}(S) \geq \frac{\rho'(S)}{1 + 6\varepsilon}.$$

By the feasibility of the problem $(G, \mathcal{D}, s)$ with demands $\tilde{\rho}$, we have $|\delta^-(S)| \geq \tilde{\rho}(S)$, and hence

$$(1+6\varepsilon)\left|\delta^-(S)\right| \geq (1+6\varepsilon)\tilde{\rho}(S) \geq \rho'(S).$$

(ii) At least a $\frac{6\varepsilon}{1+6\varepsilon}$ fraction of the total ρ'–demand of S is associated with leaders of split arborescences. Since the maximum demand contained in any single arborescence is at most ε, there are at least

$$\frac{6\varepsilon}{(1+6\varepsilon)} \cdot \frac{\rho'(S)}{\varepsilon} = \frac{6\rho'(S)}{1+6\varepsilon}$$

split arborescences in S. Since these arborescences are half–disjoint, it follows that

$$\left|A \cap \delta^+(S)\right| \geq \frac{3\rho'(S)}{1+6\varepsilon}.$$

We now have two sub–cases.

(a) At least a $\frac{2}{3}$ fraction of the edges in $A \cap \delta^+(S)$ are dense. Thus, a total flow value of at least $(1+6\varepsilon)^{-1}\rho'(S)$, in the problem $(G, \mathcal{D}, s)$, is leaving the set S, and consequently at least this much flow is entering the set. But since edges have unit capacity in $(G, \mathcal{D}, s)$, we must have $|\delta^-(S)| \geq (1+6\varepsilon)^{-1}\rho'(S)$. Thus,

$$(1+6\varepsilon)\left|\delta^-(S)\right| \geq \rho'(S).$$

(b) At least a $\frac{1}{3}$ fraction of the edges in $A \cap \delta^+(S)$ are shallow. By the definition of A, all these edges have their heads in shallow nodes, $z_1, \ldots, z_r$. Because we are considering a violated cut whose cardinality is maximal, it must be that each z_i is also a branching node — for otherwise, $S \cup \{z_i\}$ would be a violated cut of larger cardinality. Thus $\{z_1, \ldots, z_r\} \subset Z$, and hence each of $P_{z_1}, \ldots, P_{z_r}$ has an edge in $\delta^+(S)$. But from this it follows that each of $P_{z_1}, \ldots, P_{z_r}$ must also have an edge in $\delta^-(S)$. Since

$$r \geq \frac{1}{3} \cdot \frac{3\rho'(S)}{1+6\varepsilon},$$

we have

$$(1+6\varepsilon)\left|\delta^-(S)\right| \geq \rho'(S).$$

Thus in all cases, we have $(1+6\varepsilon)|\delta^-(S)| \geq \rho'(S)$, contradicting our assumption that S was a set violating the cut condition. Hence there are no violated cuts in $(G, U(\vec{\Gamma}), s)$, and so it is fractionally feasible with edge capacities equal to $1+6\varepsilon$. ∎

6.2 The Approximation Algorithms

Using Lemma 6.8 in place of Lemmas 2.3 and 2.4, we will be able to obtain approximation algorithms for the three basic objective functions; the constants will be somewhat larger due to the weaker bound provided by Lemma 6.8.

The only essential difference between Lemmas 2.4 and 6.8, in terms of their application, is the following. The coalition produced by Lemma 6.8 depends on the structure of $(G, \mathcal{D}, s)$, and in particular on the demand associated with each terminal in $\mathcal{D}$. This issue did not arise in the undirected case, since Lemma 2.4 applied to any coalition.

Congestion

Theorem 6.9 *Let G be directed and $(G, \mathcal{D}, s)$ be fractionally feasible. Then it has an unsplittable routing of relative congestion at most $1 + 9\sqrt{\rho^*} + 9\rho^* = (1 + 3\sqrt{\rho^*})^2$.*

Proof. We set $\sigma_0 = \frac{1}{3}\sqrt{\rho^*}$, and use Lemma 6.8 to construct a coalition $\vec{\Gamma}$ with respect to the demands $\{\rho_j\}$ on the terminals in $\mathcal{D}$. Now, following the proof of Theorem 3.1, we multiply all capacities in G by $1 + 6\sigma_0$ and then round each capacity up to the nearest multiple of σ_0. By Lemma 6.8, there is a feasible fractional routing of s to the leaders of $\vec{\Gamma}$ in the resulting graph. Thus, by Theorem 6.1, there is a feasible unsplittable routing of s to the leaders of $\vec{\Gamma}$, using paths $\{Q_i\}$.

Again following the proof of Theorem 3.1, if $t_j \in \mathcal{D}$ belongs to $\tau^{-1}(A_i)$, then we concatenate the path Q_i with the unique path u_i-t_j path in the arborescence A_i; we use R_j to denote this path. This produces an unsplittable flow path for each terminal $t_j \in \mathcal{D}$. The total relative congestion due to the Q-parts of all these flow paths is at most

$$\frac{\sigma_0 + \rho^*}{\sigma_0} \cdot (1 + 7\sigma_0),$$

and the total relative congestion due to the R-parts of all these flow paths is at most

$$2(\sigma_0 + \rho^*) = \frac{\sigma_0 + \rho^*}{\sigma_0} \cdot (2\sigma_0).$$

Adding these two bounds, we obtain a total relative congestion of at most

$$\frac{\sigma_0 + \rho^*}{\sigma_0} \cdot (1 + 9\sigma_0) = (1 + 3\sqrt{\rho^*})^2.$$

■

When $\rho^* = 1$, this gives a bound of 16. Again, it would be interesting to determine whether the correct bound could be as low as 2.

From this theorem, as in Section 3, we obtain the following corollaries.

Corollary 6.10 *Let G be directed. Suppose that in $(G, \mathcal{D}, s)$ we have $\rho^* < \frac{1}{9}$, and the cut condition is met by a factor of at least $\beta = (1 - 3\sqrt{\rho^*})^{-2}$. Then $(G, \mathcal{D}, s)$ has a feasible unsplittable solution.*

Corollary 6.11 *There is a 16–approximation to $\nu(G, \mathcal{D}, s)$ in an arbitrary directed graph G.*

Maximization

We first observe that, using Theorem 6.1 in place of Theorem 1.1, Lemmas 4.1 and 4.2 hold in the directed case. The directed analogue of Theorem 4.3 is the following.

Theorem 6.12 *Let G be a directed graph and $\rho^* < \frac{11}{49} - \frac{6\sqrt{2}}{49}$. There is a polynomial–time algorithm for the single–source* UFP *producing a flow of value at least $1 - 6\sqrt{\rho^*} + 7\rho^*$ times $\alpha^f(G, \mathcal{D}, s)$.*

Proof. Set $\sigma_0 = \frac{\sqrt{\rho^*}}{3-7\sqrt{\rho^*}}$. We require $\rho^* < \frac{11}{49} - \frac{6\sqrt{2}}{49}$ so that the expression $1 - 6\sqrt{\rho^*} + 7\rho^*$ is positive and σ_0 is well–defined.

We compute a fractional maximum flow in $(G, \mathcal{D}, s)$, and let ε_j denote the demand routed to the terminal $t_j \in \mathcal{D}$. Using the set of demand values $\{\varepsilon_j\}$ for the terminals in $\mathcal{D}$, we apply Lemma 6.8 to produce a strong coalition $\vec{\Gamma}$ of order σ_0.

Let G' denote the graph obtained by multiplying all capacities of G by $1 + 6\sigma_0$ and then rounding up to the nearest multiple of σ_0. We assign each non–exceptional vertex in $U(\vec{\Gamma})$ a demand of σ_0, and each exceptional vertex in $U(\Gamma)$ a demand equal to the amount assigned to its arborescence. As in the proof of Theorem 4.3, there is a maximum flow in $(G', U(\vec{\Gamma}), s)$ that is an unsplittable flow. Moreover, by Lemma 6.8, if we were to assign to $u_i \in U(\vec{\Gamma})$ a demand equal to

$$\sum_{j:t_j \in A_i} \varepsilon_j,$$

we would obtain a fractionally feasible flow problem. It follows that the maximum flow value in $(G', U(\vec{\Gamma}), s)$ is at least as large as the fractional maximum flow value in $(G, \mathcal{D}, s)$.

Now, still following the proof of Theorem 4.3, we take each $u_i \in U(\vec{\Gamma})$ that is routed by this optimum unsplittable flow and route a subset of the terminals in its arborescence of total demand at least

$$\left(\frac{1 - 2\sigma_0}{1 + 7\sigma_0}\right) \sigma_0 - \rho^*.$$

One argues as before that the set of flow paths for all these routed terminals satisfies the capacity constraints, and we therefore calculate that we have routed a total demand that is within a factor of

$$\frac{\left(\frac{1-2\sigma_0}{1+7\sigma_0}\right) \sigma_0 - \rho^*}{\sigma_0} = 1 - \frac{\frac{9\sqrt{\rho^*}}{3-7\sqrt{\rho^*}}}{\frac{3}{3-7\sqrt{\rho^*}}} - \frac{\rho^*}{\frac{\sqrt{\rho^*}}{3-7\sqrt{\rho^*}}} = 1 - 6\sqrt{\rho^*} + 7\rho^*. \tag{11}$$

of the fractional optimum in $(G, \mathcal{D}, s)$. ■

Corollary 6.13 *There is a constant–factor approximation algorithm for $\alpha(G, \mathcal{D}, s)$ in an arbitrary directed graph G.*

Routing in Rounds

Theorem 5.2, that the realizable subsets in an integral single–source UFP form a matroid, holds in the directed case as well. Consequently, Lemmas 5.4, 5.5, and 5.6 hold in the directed case. Finally, the analogue of Theorem 5.7 is the following; the proof is again a direct modification of the undirected version.

Theorem 6.14 *Let $\rho^* < \frac{29}{98} - \frac{3\sqrt{22}}{49}$. There is a polynomial–time algorithm that routes $(G, \mathcal{D}, s)$ in $2\lceil \nu_0^f(G, \mathcal{D}, s)\rceil$ rounds.*

Corollary 6.15 *Let $(G, \mathcal{D}, s)$ be an arbitrary single–source unsplittable flow problem, with G a directed graph. There is a polynomial–time constant–factor approximation for $\chi(G, \mathcal{D}, s)$.*

Corollary 6.16 *Let G be directed. If $(G, \mathcal{D}, s)$ is fractionally feasible, then it is routable without splitting in a constant number of rounds. If additionally $\rho^* < \frac{29}{98} - \frac{3\sqrt{22}}{49}$, then it is routable without splitting in two rounds.*

7 Minimum-Cost Unsplittable Flow

We now consider an "unsplittable" version of the well-known minimum cost flow problem. We are given a graph with both capicities and *costs* on the edges, and we want to find an unsplittable flow whose congestion and cost are both small.

Thus, in this section, let $G = (V, E)$ denote an undirected graph, $s \in V$ a *source vertex*, and $\mathcal{D} = \{t_1, \ldots, t_k\} \subset V$ a set of *terminals* with demands $\rho_1, \ldots, \rho_k \in (0, 1]$. As before, let $\rho^* = \max_i \rho_i$. Associated with each edge $e \in E$ is a capacity $c_e \geq 1$ and a *cost* $\gamma_e \geq 0$. If P is a path in G, we define the cost of P, $\gamma(P)$, to be the sum of the cost of its edges; and if u and v are two vertices of G, we define the *weighted distance* between them, $\gamma(u, v)$, to be the minimum cost of a path between them. Given a routing of $(G, \mathcal{D}, s)$, consisting of paths $P_1, \ldots, P_k$, we define the cost of the routing to be the sum $\sum_i \rho_i \gamma(P_i)$.

We wish to prove the following extension of Theorem 3.1, in the setting of minimum-cost unsplittable flow.

Theorem 7.1 *Let $(G, \mathcal{D}, s)$ have a feasible fractional routing $\mathcal{F}$ of cost ξ. Then it has an unsplittable routing of relative congestion at most $1 + 2\sqrt{5\rho^*} + 5\rho^*$ and cost at most $(3 + 2\sqrt{5\rho^*})\xi$.*

Proof. We follow the construction used in the proof of Theorem 3.1, with the following modifications.

(i) We set $\sigma_0 = \sqrt{\frac{1}{5}\rho^*}$ and let Γ be a strong coalition in $(G, \mathcal{D}, s)$ of order σ_0. In the construction of the strong coalition given by the proof of Lemma 2.3, we have the freedom to begin from any spanning tree T of G. We take T to be a shortest-paths tree rooted at s with respect to the cost function γ. That is, for each $v \in V$, the unique s-v path in T is a minimum-cost s-v path.

(ii) Each tree T_i of Γ is a subtree of T, and hence it contains a unique node u_i with the property that, for all $v \in T_i$, u_i lies on the unique s-v path in T. We choose u_i to be the *leader* of T_i (regardless of whether u_i is a center node), and let $U'(\Gamma)$ denote this set of leader nodes.

We assign each $u_i \in U'(\Gamma)$ a demand of σ_0 if $u_i \neq s$, and 0 otherwise. Now, Lemma 2.4 cannot be directly applied to the routing problem $(G, U'(\Gamma), s)$, since the nodes u_i are not necessarily center nodes. However, we can still use the construction in the proof of Lemma 2.4 to build a fractional routing $\mathcal{F}'$ for $(G, U'(\Gamma), s)$ from the feasible fractional routing $\mathcal{F}$ of cost ξ in $(G, \mathcal{D}, s)$. As in the proof of that lemma, the fractional flow paths for $(G, U'(\Gamma), s)$ consist of M-parts (corresponding to the paths of the fractional flow $\mathcal{F}$ in $(G, \mathcal{D}, s)$) and N-parts (consisting of paths from u_i to each of the terminals in its tree). The M-parts of all constructed flow paths still have relative congestion at most 1. The N-parts now have relative congestion at most $2\sigma_0$, since each edge carries at most σ_0 units of flow for each of the two trees of Γ that contain it. Thus $(G, U'(\Gamma), s)$ has a fractional routing of relative congestion at most $1 + 2\sigma_0$.

Let us now compute the cost of this fractional routing $\mathcal{F}'$. First observe that the quantity

$$\sum_j \rho_j \gamma(s, t_j)$$

is a lower bound on the minimum cost of a fractional routing in $(G, \mathcal{D}, s)$, and hence is no more than ξ. Second, by the construction of the coalition Γ from a shortest-paths tree rooted at s with respect to γ, and by choice of the leaders u_i, we have $\gamma(u_i, t_j) \leq \gamma(s, t_j)$. Thus

$$\sum_j \rho_j \gamma(u_i, t_j) \leq \xi. \tag{12}$$

Now, each N-part of a path in the routing $\mathcal{F}'$ consists of a path from u_i to some $t_j \in \tau^{-1}(T_i)$, carrying at most ρ_j units of flow. Thus, by Equation (12), the total cost of all N-parts is at most ξ. Since the M-parts of all paths in $\mathcal{F}'$ correspond directly to flow paths in $\mathcal{F}$, their total cost is at most ξ as well. Thus the total cost of the fractional routing $\mathcal{F}'$ is at most 2ξ.

Let G' denote the graph obtained from G by multiplying all capacities by $1+2\sigma_0$ and then rounding each capacity up to the nearest multiple of σ_0. Note that each resulting capacity c'_e satisfies

$$c'_e \leq (1 + 2\sigma_0)c_e + \sigma_0 \leq (1 + 3\sigma_0)c_e. \tag{13}$$

By the above arguments, $(G', U'(\Gamma), s)$ has a feasible fractional routing of cost at most 2ξ. Since all demands in $(G', U'(\Gamma), s)$ are equal to 0 or σ_0, all capacities are multiples of σ_0, and there is a feasible fractional routing of cost 2ξ, the integrality theorem for minimum-cost flows implies that there is an unsplittable routing for $(G', U'(\Gamma), s)$ of cost $\xi' \leq 2\xi$. Let Q_i denote the path for $u_i \in U'(\Gamma)$ in this unsplittable routing.

Finally, we construct the unsplittable routing for $(G, \mathcal{D}, s)$ as follows. For each terminal $t_j \in \mathcal{D}$, such that $\tau(t_j) = T_i$, we define R_j to be the unique t_j-u_i path in T_i. We define the unsplittable routing for $(G, \mathcal{D}, s)$ by defining the path P_j for $t_j \in \mathcal{D}$ to be the concatenation of R_j with Q_i.

By Equation (13), the relative congestion due to the Q-parts of all paths is at most

$$\frac{\sigma_0 + \rho^*}{\sigma_0} \cdot (1 + 3\sigma_0) = 1 + \frac{3}{\sqrt{5}}\sqrt{\rho^*} + \sqrt{5\rho^*} + 3\rho^*. \tag{14}$$

By the fact that each edge is contained in at most two trees of Γ, and the fact that each tree of Γ contains a total demand of at most $\sigma_0 + \rho^*$, the relative congestion due to the R-parts of all paths is at most

$$2\sigma_0 + 2\rho^* = \frac{2}{\sqrt{5}}\sqrt{\rho^*} + 2\rho^* \tag{15}$$

Adding (14) and (15), the relative congestion on any edge is at most

$$1 + 2\sqrt{5\rho^*} + 5\rho^*,$$

as claimed.

We bound the cost of this unsplittable routing as follows. The total cost of the Q-parts of all paths is at most

$$\frac{\sigma_0 + \rho^*}{\sigma_0} \cdot \xi' = (1 + \sqrt{5\rho^*})\xi' \leq (2 + 2\sqrt{5\rho^*})\xi.$$

The total cost of the R-parts of all paths is, by Equation (12), at most ξ. Thus the total cost of the unsplittable routing is at most $(3 + 2\sqrt{5\rho^*})\xi$. ■

8 Unsplittable Flow in Densely Embedded Graphs

For the sake of concreteness, we focus on a particular densely embedded graph, namely the two–dimensional mesh G. Our goal is to give a constant–factor approximation for the general unsplittable flow maximization problem in G, in which all demands are at most 1, and all edges have capacity 1. The result will essentially be obtained by incorporating the algorithm of Corollary 3.2 into a disjoint paths approximation algorithm due to the author and Éva Tardos [16] We assume a fair amount of familiarity with the arguments in [16].

As in that paper, we consider *long* and *short* connections separately. Short connections are handled recursively. We call a long connection (s_i, t_i) *heavy* if $\rho_i \geq \frac{1}{16}$ and *light* otherwise. Since the approximation ratio for long connections in [16] holds relative to the fractional optimum, we can run the basic algorithm of [16] on the heavy connections, assuming that all of them have demand 1. This loses only a constant factor in the approximation ratio.

Hence we are left with the problem of routing the long, light connections. Following the algorithm of [16], we cover the mesh with a set of *clusters* $\{C_i\}$ and construct a *simulated network* $\mathcal{N}$ on this set of clusters. We use an algorithm of Raghavan [22] to produce "global routes" in this network. As in [16], these can be converted into paths in the mesh; if (s_i, t_i) is a terminal pair, then the resulting path has its ends on the boundaries of the clusters containing s_i and t_i.

Thus, what we need is to develop an analogue of Lemma 3.14 of [16], to deal with the "unsplittable escape problem" that arises in trying to route the terminals contained in a cluster C_i to the boundary of C_i. We do this as follows. After the rounding step of Raghavan's algorithm, we say that a rectangular set R of nodes in C_i is *overfull* if the amount of demand in R is more than $\frac{1}{4}$ times as large as $|\delta(R)|$. We delete any terminal pair with one end in an overfull rectangle. As in the argument from [16], there is a constant probability that a given terminal pair is not deleted, if we set the scaling factor small enough in Raghavan's algorithm.

Following this deletion step, the cut condition for all rectangular subsets of C_i is met by a factor of 4. As in [16], this implies that the cut condition for all subsets of C_i is met by a factor 4. Since the maximum demand of a terminal in C_i is at most $\frac{1}{16}$, Corollary 3.2 of the present paper implies that the set of terminals in C_i that have not been deleted can be routed on unsplittable paths to the boundary of C_i, as desired.

Thus we have

Theorem 8.1 *There is a constant–factor approximation for the general unsplittable flow maximization problem in the uniform–capacity two–dimensional mesh.*

One can also obtain a constant–factor approximation for $\nu(\mathcal{T})$ in densely embedded graphs, using Corollary 3.3 together with an algorithm of the author, Satish Rao, and Éva Tardos. Details will appear in [17].

Acknowledgements

We thank Éva Tardos for valuable discussions on these problems. We thank Michel Goemans for suggesting the use of center nodes as a way to improve the constants, and for suggesting the proof of Lemma 2.4 that we have presented in this version of the paper. The overall presentation also benefited from his thorough reading of the paper. Finally, we thank Serge Plotkin for suggesting that it would be of interest to consider the version of the problem with costs on the edges.

References

[1] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin and O. Waarts, "On-line load balancing with applications to machine scheduling and virtual circuit routing" *Proc 25th ACM Symp. on Theory of Computing*, 1993. pp. 623-631.

[2] Y. Aumann, Y. Rabani, "Improved bounds for all-optical routing," *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms*, 1995, pp. 567–576.

[3] B. Awerbuch, Y. Azar, S. Plotkin, "Throughput-competitive online routing," *Proc. 34th IEEE Symp. on Foundations of Computer Science*, 1993, pp. 32–40.

[4] B. Awerbuch, R. Gawlick, F.T. Leighton, Y. Rabani, "On-line admission control and circuit routing for high performance computing and communication," *Proc. 35th IEEE Symp. on Foundations of Computer Science*, 1994, pp. 412–423.

[5] S. Cosares, I. Saniee, "An optimization problem related to balancing loads on SONET rings," *Telecommunications Systems*, 3(1994), pp. 165–181.

[6] J. Edmonds, "Minimum partition of a matroid into independent subsets,", *J. Res. Nat. Bur. Standards B*, 69(1965), pp. 67–72.

[7] L.R. Ford, D.R. Fulkerson, "Maximal flow through a network," *Canadian J. Math*, 8(1956), pp. 399–404.

[8] L.R. Ford, D.R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[9] A. Frank, "Packing paths, cuts, and circuits — a survey," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990, pp. 49–100.

[10] R. Gawlick, C. Kalmanek, K.G. Ramakrishnan, "On–line routing for permanent virtual circuits," *Proc. INFOCOM*, 1995, pp. 278–288.

[11] N. Garg, V. Vazirani, M. Yannakakis, "Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover," *Proc. International Colloq. on Automata, Languages, and Programming*, 1993, pp. 64–75.

[12] D. Hochbaum, D. Shmoys, "Using dual approximation algorithms for scheduling problems: theoretical and practical results," *Journal of the ACM*, 34(1987), pp. 144–162.

[13] R.M. Karp, "On the computational complexity of combinatorial problems," *Networks* 5(1975), pp. 43–68.

[14] R.M. Karp, F.T. Leighton, R. Rivest, C. Thompson, U. Vazirani, V. Vazirani, "Global wire routing in two–dimensional arrays," *Algorithmica*, 2(1987), pp. 113–129.

[15] J. Kleinberg, É. Tardos, "Approximations for the Disjoint Paths Problem in High–Diameter Planar Networks," *Proc. 27th ACM Symp. on Theory of Computing*, 1995, pp. 26–35.

[16] J. Kleinberg, É. Tardos, "Disjoint Paths in Densely Embedded Graphs," *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 1995, pp. 52–61.

[17] J. Kleinberg, S. Rao, É. Tardos, "Routing with low congestion in densely embedded graphs," manuscript.

[18] J.K. Lenstra, D.B. Shmoys, É. Tardos, "Approximation algorithms for scheduling unrelated parallel machines," *Proc. 28th IEEE Symp. on Foundations of Computer Science*, 1987.

[19] K. Menger, "Zur allgemeinen Kurventheorie," *Fundam. Math.*, 19(1927), pp. 96–115.

[20] H. Perfect, "Applications of Menger's graph theorem," *J. Math. Analysis Appl.* 22(1968), pp. 96–111.

[21] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood Ltd., 1993.

[22] P. Raghavan, "Probabilistic construction of deterministic algorithms: approximating packing integer programs," *J. Computer and System Sciences*, 37(1988), pp. 130–143.

[23] P. Raghavan, C.D. Thompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, 7(1987), pp. 365–374.

[24] P. Raghavan, E. Upfal, "Efficient all-optical routing," *Proc. 26th ACM Symp. on Theory of Computing*, 1994, pp. 134–143.

[25] N. Robertson, P.D. Seymour, "Outline of a disjoint paths algorithm," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990

[26] A. Schrijver, "Homotopic routing methods," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990.

[27] A. Schrijver, P.D. Seymour, P. Winkler, "The ring loading problem," submitted for publication.

[28] D.J.A. Welsh, *Matroid Theory*, London: Academic Press, 1976.