Edge-disjoint Paths and Unsplittable Flow

Stavros G. Kolliopoulos

Department of Informatics and Telecommunications

National and Kapodistrian University of Athens

Athens 157 84, Greece

E-mail: sgk@di.uoa.gr

December 1, 2006

1 Introduction

Finding disjoint paths in graphs is a problem that has attracted considerable attention from at least three perspectives: graph theory, VLSI design and network routing/flow. The corresponding literature is extensive. In this chapter we limit ourselves mostly to results on offline approximation algorithms for problems on general graphs as influenced from the network flow perspective. Surveys examining the underlying graph theory, combinatorial problems in VLSI, and disjoint paths on special graph classes can be found in [35, 36, 86, 88, 76, 83, 75, 51].

An instance of *disjoint paths* consists of a (directed or undirected) graph $G = (V, E)$ and a multiset $\mathcal{T} = \{(s_i, t_i) : s_i \in V, t_i \in V,\ i = 1, \ldots, k\}$ of k source-sink pairs. Any source or sink is called a *terminal.* An element of $\mathcal{T}$ is also called a *commodity.* One seeks a set of edge- (or vertex-)disjoint paths $P_1, P_2, \ldots, P_k$, where P_i is an $s_i - t_i$ path, $i = 1, \ldots, k$. In the case of vertex-disjoint paths we are interested in paths that are internally disjoint, i.e., a terminal may appear in more than one pair in $\mathcal{T}$. We abbreviate the edge-disjoint paths problem by EDP. The notation introduced will be used throughout the chapter to refer to an input instance. We will also denote $|V|$ by n and $|E|$ by m for the corresponding graph.

Based on whether G is directed or undirected and the edge- or vertex-disjointness condition one obtains four basic problem versions. The following polynomial-time reductions exist among them. Any undirected problem can be reduced to its directed counterpart by replacing an undirected edge with an appropriate gadget; both reductions maintain planarity. See [78] and [88, Chapter 70] for details. An edge-disjoint problem can be reduced to its vertex-disjoint counterpart by replacing G with its line graph (or digraph as the case may be). Directed vertex-disjoint paths reduce to directed edge-disjoint paths by replacing every vertex with a pair of new vertices connected by an edge. There is no known reduction from a directed to an undirected problem. The reader should bear in mind these transformations throughout the chapter. They can serve for translating approximation guarantees or hardness results from the edge-disjoint to the vertex-disjoint setting and vice versa.

The *unsplittable flow* problem (UFP) is the generalization of EDP where every edge $e \in E$ has a positive capacity u_e, and every commodity i has a demand $d_i > 0$. The demand from s_i to t_i has to be routed in an unsplittable manner, i.e., along a single path from s_i to t_i. For every edge e the total demand routed through that edge should be at most u_e. We will often refer to a capacitated graph as a *network.* In a similar manner a vertex-capacitated generalization of vertex-disjoint paths can be defined. UFP was introduced in the PhD thesis of Kleinberg [51]. Versions of the problem had been studied before though not under the UFP moniker (see, e.g., [22, 6]).

If one relaxes the requirement that every commodity should use exactly one path, one obtains the *multicommodity flow* problem which is well known to be solvable in polynomial time. When all the sources of a multicommodity flow instance coincide at a vertex s and all the sinks at a vertex t, we obtain the classical *maximum flow* problem to which we also refer to as *s-t flow.* The relation between UFP and multicommodity flow is an important one to which we shall return often in this survey. We will denote a solution to either problem as a flow vector f, defined on the edges or the paths of G as appropriate.

1.1 Complexity of disjoint-path problems. For general k all four basic problems are NP-complete. The undirected vertex-disjoint paths problem was shown to be NP-complete by Knuth in 1974 (see [49]), via a reduction from SAT, and by Lynch [71]. This implies the NP-completeness of directed vertex-disjoint paths and directed edge-disjoint paths. Even, Itai and Shamir [30] showed that both problems remain NP-complete on directed acyclic graphs (DAGs). In the same paper the undirected edge-disjoint paths problem was shown NP-complete even when $\mathcal{T}$ contains only of two distinct pairs of terminals. In the case when $s_1 = s_2 = \ldots = s_k$ all four versions are in P as special cases of maximum flow. For planar graphs Lynch's reduction [71] shows NP-completeness for undirected vertex-disjoint paths; Kramer and van Leeuwen [65] show that undirected EDP is NP-complete. The NP-completeness of the directed planar versions follows.

For fixed k, the directed versions are NP-complete even for the case of two pairs with opposing source-sinks, i.e., (s,t) and (t,s) [34][1]. Undirected vertex-disjoint paths, and by implication edge-disjoint paths as well, can be solved in polynomial time [84]. This is an outcome of the celebrated project of Robertson and Seymour on graph minors. See [13] for an informal description of the highly impractical Robertson-Seymour algorithm. It is notable that for fixed k, vertex-disjoint paths, and by consequence EDP, can be solved on DAGs by a fairly simple polynomial-time algorithm [34]. Earlier polynomial-time algorithms for $k = 2$ include the one by Perl and Shiloach on DAGs [78] and the ones derived independently by Seymour [89], Shiloach [91] and Thomassen [102] for vertex-disjoint paths on general undirected graphs.

For planar graphs and fixed k the directed vertex-disjoint path problem is in P [87] while the complexity of the edge-disjoint case is open. When the input graph is a tree, Garg, Vazirani and Yannakakis gave a polynomial-time algorithm to maximize the number of pairs that can be connected by edge-disjoint paths [41]. The algorithm extends for vertex-disjoint paths [N. Garg, personal communication, July 2005]. By total unimodularity, the EDP maximization problem is polynomial-time solvable on *di-trees* as well, i.e., directed graphs in which there is a unique directed path from s_i to t_i, for all i; a reduction to a minimum-cost circulation problem is also possible in this case (cf. [23]). Reducing directed vertex-disjoint paths to EDP maintains the di-tree property, hence the former problem is in P as well. Observe that directed out- and in-trees are special cases of di-trees.

Our presentation will focus mostly on EDP and its generalization to edge-capacitated UFP . We will switch explicitly to vertex-disjoint paths when necessary. Approximation algorithms for vertex-disjoint paths are typically obtained by modifying appropriately algorithms for the edge-disjoint case.

[1] The NP-completeness proof holds for a sparse graph with $m = \Theta(n)$; this observation has consequences for hardness of approximation proofs in [44, 7]

1.2 Optimization versions. Two basic NP-hard optimization problems are associated with unsplittable flow and hence with EDP. Given a UFP instance an *unsplittable flow solution* or simply a *routing* is a selection of $k' \leq k$ paths, one each for a subset $\mathcal{T}' \subseteq \mathcal{T}$ of k' commodities. Any routing can be expressed as a flow vector f; the flow f_e through edge e equals the sum of the demands using e. A *feasible* routing is one that respects the capacity constraints. In the *maximum demand* optimization problem one seeks a feasible routing of a subset $\mathcal{T}'$ of commodities such that $\sum_{i \in \mathcal{T}'} d_i$ is maximized. The *congestion* of a routing f is defined as $\max_{e \in E}\{\max\{f_e/u_e, 1\}\}$. Note that the events $f_e < u_e$ and $f_e = u_e$ are equivalent for this definition. In the *minimum congestion* optimization problem one seeks a routing of all k commodities that minimizes the congestion, i.e., one seeks the minimum $\lambda \geq 1$ such that all k commodities can be feasibly routed if all the capacities are multiplied by λ. From now on, when we refer to EDP without further qualification we imply the maximum demand version of EDP. Some other objective functions of interest will be defined in Section 3.

1.3 Main threads. We present now some of the unifying themes in the literature on approximation algorithms for disjoint-path problems.

LP-rounding algorithms. As mentioned above, multicommodity flow is an efficiently-solvable relaxation of EDP. Hence it is no accident that multicommodity flow theory has played such an important part in developing algorithms for disjoint-path problems. This brings us to the standard linear programming formulation for multicommodity flow. Let $\mathcal{P}_i$ denote the set of paths from s_i to t_i. Set $\mathcal{P} := \bigcup_{i=1}^{k} \mathcal{P}_i$. Consider the following linear program (LP) for maximum multicommodity flow:

$$
\begin{aligned}
\text{maximize} \sum_{P \in \mathcal{P}} f_P & & \text{(LP-MCF)} \\
\sum_{P \in \mathcal{P}_i} f_P \leq d_i & \quad \text{for } i = 1, \ldots, k & \\
\sum_{P \in \mathcal{P}\,:\,P \ni e} f_P \leq u_e & \quad \text{for } e \in E & \\
f_P \geq 0 & \quad \text{for } P \in \mathcal{P} &
\end{aligned}
$$

The number of variables in the LP is exponential in the size of the graph. By using flow variables defined on the edges one can write an equivalent LP of polynomial size. We choose to deal with the more elegant flow-path formulation. Observe that adding the constraint $f_P \in \{0, d_i\}, \forall P \in \mathcal{P}_i$, to (LP-MCF) turns it into an exact formulation for maximum demand Ufp. A similar LP, corresponding to the *concurrent flow* problem, can be written for minimizing congestion. See [104] for details. We call an LP solution for the optimization problem of interest *fractional.* Several early approximation algorithms for Ufp, and more generally integer multicommodity flow, work in two stages. First a fractional solution f is computed. Then f is rounded to an unsplittable solution $\hat{f}$ through procedures of varying intricacy, most commonly by randomized rounding as shown by Raghavan and Thompson [82]. The randomized rounding stage can usually be derandomized using the method of conditional probabilities [28, 95, 81]. The derandomization component has gradually become very important in the literature for two reasons. First, through the key work of Srinivasan [98, 96] on pessimistic estimators, good deterministic approximation algorithms were designed even in cases where the success probability of the randomized experiment was small. See [97, 11] for applications to disjoint paths. Second, in some cases the above two-stage scheme can be implemented rather surprisingly without solving first the linear program. Instead one designs directly a suitable Langrangean relaxation algorithm implementing the derandomization part. See the work of Young [105] and Chapter R-2 in this volume.

We note that some of the approximation ratios obtained through the LP-rounding method can nowadays be matched (or surpassed) by simple combinatorial algorithms. By combinatorial one usually means algorithms restricted to ordered ring operations as opposed to ordered field ones. Two distinct greedy algorithms for Edp were given by Kleinberg [51] (see also [55]), and Kolliopoulos and Stein [62] (see also [57]). Most of the subsequent work on combinatorial algorithms uses these two approaches as a basis. Still the influence of rounding methods on the development of algorithms for disjoint-path problems can hardly be overstated. See Chapters 6 and 7 in this volume for further background on LP-based approximation algorithms.

Approximate max-flow min-multicut theorems. One of the first results on disjoint paths and in fact one of the cornerstones of graph theory is Menger's Theorem [74]: an undirected graph is k vertex-connected if and only if there are k vertex-disjoint paths between any two vertices. The edge analogue holds as well and the min-max relation behind the theorem has resurfaced in a number of guises, most notably as the max-flow min-cut theorem for s-t flows. Let $G = (V, E)$ be undirected. For $U \subseteq V$, define $\delta(U) := \{\{u, v\} \in E : u \in U \text{ and } v \in V \setminus U\}$. Similarly $dem(U)$ is the sum of all demands over commodities which are separated by the cut $\delta(U)$. A necessary condition for the existence of a feasible fractional solution to (LP-MCF) that satisfies all demands is the *cut condition:*

$$\sum_{e\in\delta(U)} u_e \geq dem(U), \text{ for each } U \subseteq V.$$

For s-t flows, the max-flow min-cut theorem [33, 38, 27] states that the cut condition is sufficient. For undirected multicommodity flow, Hu showed that the cut condition is sufficient for $k = 2$ [45]. It fails in general for $k \geq 3$. For directed multicommodity flows there are simple examples with $k = 2$, for which the directed analogue of the cut condition holds but the demands cannot be satisfied fractionally (see, e.g., [88]). For undirected EDP, already for $k = 2$ the cut condition is not sufficient for a solution to exist [35].

Starting with the seminal work of Leighton and Rao [68] a lot of effort has been spent on establishing approximate multicommodity max-flow min-cut theorems. A *multicut* in an undirected graph $G = (V, E)$ is a subset of edges $F \subseteq E$, such that if all edges in F are deleted none of the pairs (s_i, t_i) $i = 1, \ldots, k$ are in the same connected component of the remaining graph. Garg, Vazirani and Yannakakis [40] showed constructively that the minimum multicut is always at most $O(\log k)$ times the maximum multicommodity flow and this is existentially tight. See [92, 104, 24] for surveys of the many results in this area and their applications to approximation algorithms. Most of this work focused originally on fractional flows. The methods were versatile enough to extend to UFP, typically yielding results that were also obtainable via randomized rounding. See, e.g., the discussion on the high-capacity UFP in Par. 3.1 below. Moreover this body of work contributed significantly to the intellectual climate that spawned, among other currents in approximation algorithms, the renewed interest in disjoint paths. This research produced also increased interest in the efficient solution of multicommodity flow problems via combinatorial approximation schemes, thereby producing fast algorithms for solving disjoint-path relaxations. Such approximation schemes had been first investigated by Shahrokhi and Matula [90]. The running time was significantly improved in [50] with extensions and refinements following in [67, 42, 80]. Extensions of these methods to general fractional packing/covering problems were first pursued in [43, 79]. A representative sample of subsequent work on fractional multicommodity flow and related problems can be found in [105, 39, 32, 47, 12].

Finally, one should acknowledge the influence of the PhD thesis of Kleinberg [51] on solidifying the various strands of work on disjoint-path problems up to the mid 90s. The results in [51] gave impetus to new research and the thesis itself is a valuable reference tool for earlier work.

The outline of this chapter is as follows. In Section 2 we present hardness of approximation results and (mostly greedy) algorithms for Edp. In Section 3 we examine the more general Ufp problem, properties of the fractional relaxation and packing integer programs. Finally in Section 4 we present results on some variants of the basic problems. Unless mentioned otherwise, all of the approximation algorithms we will describe in the upcoming sections work equally well on directed and undirected graphs.

2 Algorithms for edge-disjoint paths

In this section we examine the problem of finding a maximum-size set of edge-disjoint paths, mostly from the perspective of combinatorial algorithms. We defer the discussion of the LP-rounding algorithms and the integrality gaps of the linear relaxations until Section 3, where we examine them in the more general context of Ufp. Similarly for some key results on expander graphs and hardness bounds particular to Ufp.

2.1 Hardness results. Guruswami et al. [44] showed that on directed graphs it is NP-hard to obtain an $O(n^{1/2-\varepsilon})$ approximation for any fixed $\varepsilon > 0$. They gave a gap-inducing reduction from the two-pair decision problem to Edp on a sparse graph with $\Theta(n)$ edges. Since this Edp problem reduces to a vertex-disjoint path instance on a graph with $N = \Theta(n)$ vertices, we obtain that is NP-hard to approximate vertex-disjoint paths on graphs with N vertices within $O(N^{1/2-\varepsilon})$, for any fixed $\varepsilon > 0$. Ma and Wang [72] showed via the PCP theorem that it is NP-hard to approximate Edp on directed graphs within $O(2^{\log^{1-\varepsilon} n})$, even when the graph is acyclic. See Chapter R-14 in this volume for background on the PCP theorem and the theory of inapproximability. For the undirected edge-disjoint path problem Andrews and Zhang [4] showed that there is no $O(\log^{1/3-\varepsilon} n)$ approximation algorithm unless $NP \subseteq ZPTIME(n^{polylog(n)})$. $ZPTIME(n^{polylog(n)})$ is the set of languages that have randomized algorithms that always give the correct answer in expected running time $n^{polylog(n)}$. The lower bound was improved to $\Omega(\log^{1/2-\varepsilon} n)$ in [2], under the same complexity-theoretic assumption. Even when congestion $C > 1$ is allowed, [2] shows that the maximization version is $\log^{\Omega(1/C)} n$-hard to approximate. Edp on undirected graphs was shown MAX SNP-hard in [41].

2.2 Greedy algorithms. The first approximation algorithm analyzed in the literature for Edp on general graphs seems to be the online *Bounded Greedy Algorithm (BGA)* in the PhD thesis of Kleinberg [51]; see also [55]. The algorithm is parameterized by a quantity L. The terminal pairs are examined in one pass. When (s_i, t_i) is considered, check if s_i can be connected to t_i by a path of length at most L. If so, route (s_i, t_i) on such a path P_i. Delete P_i from G and iterate. To simplify the analysis we assume that the last terminal pair is always routed if all the previous pairs have been rejected.

The idea behind the analysis of BGA [51] is very simple but it has influenced later work such as [62], [64] and [15]. Informally it states that *in any graph there cannot be too many long paths that are edge-disjoint.* In [51] the algorithm was shown to achieve a $(2L+1)$-approximation if $L = \max\{diam(G), \sqrt{m}\}$. Several people quickly realized that the analysis can be slightly altered to obtain an $O(\sqrt{m})$-approximation. We provide such an analysis with $L = \sqrt{m}$. The first published $O(\sqrt{m})$ approximation for EDP was given by Srinivasan using LP-rounding methods [97].

Let $\mathcal{O}$ be a maximum-cardinality set of edge-disjoint paths connecting pairs of $\mathcal{T}$. Let $\mathcal{B}$ be the set of paths output by BGA and $\mathcal{O}_u \subset \mathcal{O}$ be the set of paths corresponding to terminal pairs unrouted by the BGA. We have that

$$|\mathcal{O}| - |\mathcal{O}_u| = |\mathcal{O} \setminus \mathcal{O}_u| \leq |\mathcal{B}|. \tag{1.1}$$

One tries to relate $|\mathcal{O}_u|$ to $|\mathcal{B}|$. This is done by observing that a commodity l routed in $\mathcal{O}_u$ was not routed in $\mathcal{B}$ because one of two things happened: (i) no path of length shorter than L exists or (ii) the existing paths from s_l to t_l were blocked by (intersect on at least one edge with) paths selected earlier in $\mathcal{B}$. The paths in $\mathcal{O}_u$ can thus be partitioned into the two corresponding subsets $\mathcal{O}_1$ and $\mathcal{O}_2$. $\mathcal{O}_1$ contains paths blocked by a path in $\mathcal{B}$ and has size at most $L|\mathcal{B}|$, since the elements of $\mathcal{B}$ are edge-disjoint paths of length at most L. The second set $\mathcal{O}_2 := \mathcal{O}_u \setminus \mathcal{O}_1$, consists of disjoint paths longer than L, hence $|\mathcal{O}_2| < m/L$. Therefore

$$|\mathcal{O}_u| < \frac{m}{L} + L|\mathcal{B}| = \sqrt{m} + \sqrt{m}|\mathcal{B}| \leq 2\sqrt{m}|\mathcal{B}|. \tag{1.2}$$

Adding inequalities (1.1) and (1.2) yields that the BGA is an $O(\sqrt{m})$-approximation algorithm. In Par. 3.3 below we return to the performance of the BGA on expander graphs.

The astute reader has noticed that the idea used in the analysis above is an old one. It goes back to the blocking flow method of Dinitz [25] for the s-t flow problem as applied to unit-capacity networks by Even and Tarjan [31]. A *blocking flow* is a flow that cannot be augmented without rerouting. The blocking flow method iterates over the residual graph. In every iteration a blocking flow is found over the subgraph of the residual graph that contains the edges on a shortest path from s to t. At the end of an iteration the distance from s to t in the new residual graph can be shown to have increased by at least one. When the distance becomes larger than L, the number of edge-disjoint paths from s to t is $O(\min\{m/L, n^2/L^2\})$ and this bounds also the remaining number of augmentations required by the algorithm [31].

Kolliopoulos and Stein [62] made the connection with the blocking flow idea explicit and proposed the offline Greedy_Path algorithm, from now on called simply *the greedy algorithm.* The motivation behind the greedy algorithm was the following: what amount of residual flow has survived if one is never allowed to reroute the flow sent along shortest paths at a given iteration? In every iteration, greedy picks the unrouted (s_i, t_i) pair such that the length of the shortest path P_i from s_i to t_i is minimized. The pair is routed using P_i. The greedy algorithm is easily seen to achieve an $O(\sqrt{m})$-approximation [62]. Using the BGA notation and analysis from above we obtain the following. See also [15].

Lemma 1.1 *Consider the restriction of the greedy algorithm that stops as soon as the minimum shortest path length among the unrouted pairs exceeds L. The approximation guarantee is at most* $\max\{L, |\mathcal{O}_2|\}$.

The analysis in [62] used the fact that $|\mathcal{O}_2| \leq m/L$. This was extended by Chekuri and Khanna [15]:

Theorem 1.1 *[15] Using the notation defined above $|\mathcal{O}_2| = O(n^2/L^2)$ for undirected simple graphs and $|\mathcal{O}_2| = O(n^4/L^4)$ for the directed case.*

The theorem together with Lemma 1.1 and [62] yield immediately that the greedy algorithm achieves an $O(\min\{\sqrt{m}, n^{2/3}\})$-approximation for undirected EDP and an $O(\min\{\sqrt{m}, n^{4/5}\})$ for directed EDP. Varadarajan and Venkataraman [103] improved the bound for directed graphs to $O(\min\{\sqrt{m}, (n \log n)^{2/3}\})$, again for the greedy algorithm. Interestingly, their argument shows the existence of a cut of size $O((n^2/L^2)\log^2(n/L))$ that separates all terminal pairs (s_i, t_i) lying at distance L or more. This brings us almost full circle back to the Even-Tarjan bound [31] for s-t flows. The latter argument demonstrates the existence of a cut of size $O(n^2/L^2)$ when the source is at distance L or more from the sink. [15] demonstrates an infinite family of directed and undirected instances on which the approximation ratio achieved by the greedy algorithm is $\Omega(n^{2/3})$. New ideas are thus required to bring the approximation down to $O(\sqrt{n})$ which in [15] is conjectured to be possible. Chekuri, Khanna and Shepherd [18] and independently Nguyen [77] have recently obtained

$O(\sqrt{n})$-approximation algorithms for EDP on undirected graphs and DAGs.

We now sketch the proof of Theorem 1.1 for the undirected case as given by Chekuri and Khanna [15]. The theorem holds for the fractional solution as well, i.e., the value ν of the maximum fractional multicommodity flow connecting terminals at distance more than L. Call a vertex of G *high-degree* if its degree is more than $6n/L$ and *low-degree* otherwise. The total capacity incident to low-degree vertices is $O(n^2/L)$. We claim that every $s_i - t_i$ path, $(s_i, t_i) \in \mathcal{T}$, must contain at least $L/6$ of the low-degree vertices. Therefore ν, the sum of flow values over the paths used in the fractional solution, is $O(n^2/L^2)$. To prove the claim consider a breadth-first search tree rooted at s_i and let *layer* L_j be the set of vertices at distance j from s_i. We will show something stronger: there are at least $L/6$ layers among the first L consisting only of low-degree vertices. Partition the layers into blocks of three contiguous layers and let B_j denote the block made up of layers $L_{3j+1}, L_{3j+2}, L_{3j+3}$. Discard the blocks which contain at least one layer consisting entirely of low-degree vertices. If $L/6$ or more blocks are discarded, we are done. So assume that we are left with at least $L/6$ blocks. The blocks are disjoint so at least one of the remaining blocks, call it B_*, must contain at most $6n/L$ vertices. Consider a high-degree vertex in the middle layer of B_*. By the breadth-first search property all its neighbors must be within B_* itself, a contradiction. This completes the proof of Theorem 1.1.

Other guarantees for general graphs. In the original paper on the greedy algorithm, it was shown to output a solution of size $\Omega(\max\{OPT^2/m_0, OPT/d_0\})$, where OPT is the optimum, m_0 is the minimum number of edges used in an optimal solution, and d_0 is the minimum average length of the paths in an optimal solution [62]. The second bound is a straightforward consequence of the first. The first bound is obtained through a somewhat more sophisticated charging scheme for the number of paths in an optimal solution blocked by the paths in $\mathcal{B}$. In conclusion, greedy gives better results in the case where there is a “sparse” optimal solution.

2.3 Acyclic digraphs. The author observed in [57] that greedy achieves an $o(n)$-approximation if the terminal pairs are disjoint and there is an acyclic optimal solution. In particular, one can show using a result in [48] that in this case $m_0 = O(n^{3/2})$; an $O(n^{3/4})$-approximation follows. Chekuri and Khanna [15] provided an $O(\sqrt{n} \log n)$-algorithm for DAGs. The following applies to general graphs. Because of the d_0-approximation outlined earlier, one can assume without loss of generality that all shortest $s_i - t_i$ paths have length $\Omega(\sqrt{n})$. Then a counting argument shows that there is a vertex u such that at least $\Omega(OPT/\sqrt{n})$ paths in the optimal solution go through this "congested" vertex u. We guess u and concentrate on finding the maximum-size *u-solution,* to our original EDP instance: this consists only of paths going through u. Devising an $O(\log n)$-approximation algorithm of the LP-rounding variety for this special case gives the desired result. Recently, Nguyen [77] showed that an optimal u-solution is polynomial-time computable in DAGs and undirected graphs.

2.4 Vertex-disjoint paths. The greedy algorithm, with the obvious modifications, connects a set of terminal pairs of size $\Omega(\max\{OPT/\sqrt{n_0}, OPT^2/n_0, OPT/d_0\})$ [62]. Here n_0 denotes the minimum size of a set of vertices used in the optimal solution and d_0 the minimum average path length in an optimal solution. By the hardness result of [44] this result is essentially tight on directed graphs, unless $P = NP$.

3 The general unsplittable flow problem

We start with some additional definitions. We assume that a UFP instance always satisfies the *balance* (also called *no-bottleneck*) *condition:* $d_{max} := \max_{i=1,\ldots,k} d_i \le u_{min} := \min_{e \in E} u_e$, i.e., any commodity can be routed through any of the edges. This assumption is common in the literature and we will refer explicitly to an *extended* UFP instance when the balance condition is not met. In the *weighted* UFP, commodity i has an associated weight (profit) $w_i > 0$; one wants to route feasibly a subset of commodities with maximum total weight. Note that maximizing demand reduces to maximizing the weight: simply set $w_i := d_i$, $i = 1, \ldots, k$. Another objective function of interest in addition to maximizing demand and minimizing congestion is routing in the *minimum number of rounds.* A round corresponds to a set of commodities that can be routed feasibly, hence one seeks a minimum-size partition of the set of commodities into feasible unsplittable flow solutions. A *uniform capacity unsplittable flow problem* (UCUFP) is a UFP in which all edges of the input graph have the same capacity value.

3.1 Randomized rounding and UFP. Some of the approximation ratios achieved by LP-rounding that we are about to present are currently also obtainable with simple greedy algorithms. See Par. 3.3 below. Nevertheless LP-rounding algorithms are analyzed with respect to the existentially weak optima of the linear relaxations. In addition their analysis yields upper bounds on the respective integrality gaps. An implementation study comparing the actual performance of the LP-based vs. the more combinatorially-flavored algorithms would be of interest. For an in-depth survey of randomization for routing problems see [99].

Minimizing congestion. The best known algorithm for congestion is also perhaps the best known example of the randomized rounding method of Raghavan and Thompson [82]. A fractional solution f to the concurrent flow problem is computed and then one path is selected independently for every commodity from the following distribution: commodity i is assigned to path $P \in \mathcal{P}_i$ with probability f_P/d_i. An application of the Chernoff bound [20] shows that with high probability the resulting congestion is $O(\log n/\log\log n)$ times the fractional optimum. The process can be derandomized using the method of conditional probabilities [81]. Young [105] shows how to construct the derandomized algorithm without having first obtained the fractional solution.

The analysis of the performance guarantee cannot be improved. Leighton, Rao and Srinivasan [66] provide an instance on a directed graph on which a fractional solution routes at most $1/\log^c n$ flow per edge, for any constant $c > 0$, while any unsplittable solution incurs congestion $\Omega(\log n/\log\log n)$. If the unsplittable solution uses only paths with nonzero fractional flow the lower bound holds for both undirected and directed instances with optimal UFP congestion 1 [66, 73]. Chuzhoy and Naor [21] show that for directed graphs there is no $c\log\log n$-approximation for some constant c, unless $NP \subseteq DTIME(n^{O(\log\log\log n)})$. For undirected graphs, Andrews and Zhang [3] show that congestion cannot be approximated within $(\log\log m)^{1-\varepsilon}$, for any constant $\varepsilon > 0$, unless $NP \subseteq ZPTIME(n^{polylog(n)})$. Trivially, it is NP-hard to approximate congestion within better than 2 in the case of EDP; this would solve the decision problem.

Maximum demand. Srinivasan published the first $O(\sqrt{m})$-approximation for EDP and more generally UCUFP in [97]. The first non-trivial $O(\sqrt{m}\log m)$-approximation for UFP was published in the IPCO version of [62]. Simultaneously and independently, Baveja and Srinivasan refined the results in [97] to obtain an $O(\sqrt{m})$-approximation for the general UFP ; this work was published in [11]. The Baveja-Srinivasan methods extend the earlier key work of Srinivasan on LP-rounding methods for approximating Packing Integer programs [98, 96]. We outline now some of the ideas in [98, 97, 11]. The algorithm computes first a fractional solution f to the (LP-MCF) linear relaxation (cf. Par. 1.3). The rounding method has two phases. First, a randomized rounding experiment is analyzed to show that it produces with positive probability a near-optimal feasible unsplittable solution. Second, the experiment is derandomized yielding a deterministic polynomial-time algorithm for computing a feasible near-optimal solution. Let y_* be the fractional optimum.

One starts by scaling down every variable f_P by an appropriate parameter $\alpha > 1$. This is done to boost the probability that after randomized rounding all edge capacities are met. Let B_i denote the event that in the unsplittable solution, the capacity of the edge $e_i \in E$ is violated. Let B_{m+1} denote the event that the routed demand will be less than $y_*/(\beta\alpha)$, for some $\beta > 1$. The quantity $\beta\alpha$ is the targeted approximation ratio. The randomized rounding method of Raghavan and Thompson in the context of UFP works by bounding the probability of the "bad" event $\bigcup_{i=1}^{m+1} B_i$ by $\sum_{i=1}^{m+1} Pr(B_i)$. Srinivasan [97] and later Srinivasan and Baveja [11] exploit the fact that the events $\overline{B_i}$ are *positively correlated:* if it is given that a routing respects the capacities of the edges in some $S \subset E$, the conditional probability that for $e_i \in E \setminus S$, $\overline{B_i}$ occurs, is at least $Pr(\overline{B_i})$. Mathematically this is expressed via the FKG inequality due to Fortuin, Ginibre and Kasteleyn (see [1, Chapter 6]). Using the positive correlation property, Baveja and Srinivasan obtain a better upper bound on $Pr(\bigcup_{e_i \in E} B_i)$ than the naive union bound and therefore can prove the existence of an unsplittable solution while using a better, i.e., smaller, $\beta\alpha$ scaling factor than traditional randomized rounding. The second ingredient of Srinivasan's method in [98, 96] is to design an appropriate pessimistic estimator to constructively derandomize the method. Such an estimator is shown for UFP as well in [11]. The by-now standard derandomization approach of Raghavan [81] fails since it relies precisely on the probability $Pr(\bigcup_{i=1}^{m+1} B_i)$ being upper-bounded by $\sum_{i=1}^{m+1} Pr(B_i)$.

Let d denote the *dilation* of the optimal fractional solution f, i.e., the maximum number of edges on any flow-carrying path. The Baveja-Srinivasan algorithm computes a solution to weighted UFP of value

$$\Omega(\max\{(y_*)^2/m, y_*/\sqrt{m}, y_*/d\}), \tag{1.3}$$

The corresponding upper bounds on the integrality gap of (LP-MCF) follow. The analysis of [98] was simplified by Srinivasan in [100] by using randomized rounding followed by alteration. Here the outcome of the random experiment is allowed to violate some constraints. It is then altered in a greedy manner to achieve feasibility. The problem-dependent alteration step should be analyzed to quantify the potential degradation of the performance guarantee. This method was applied to UFP in [14].

For weighted **vertex-disjoint** paths the corresponding bounds hold with n in place of m [11, 62]. In addition to the upper bounds on the integrality gap of (LP-MCF) given by (1.3), the integrality gap for EDP is $O(\sqrt{n})$ on undirected graphs [18] and $O(n^{4/5})$ on directed graphs [15]. The gap is known to be at least $k/2$ by an example in a grid-like planar graph with $k = \Theta(\sqrt{n})$, even for the EDP case [41].

Minimizing the number of rounds. Aumann and Rabani [5] (see also [51]) show that a ρ-approximation for maximum demand translates to an $O(\rho \log n)$ guarantee for the number of rounds objective. [11] provides improvements when all edge capacities are unit. Let $\chi(\mathcal{T})$ be the minimum number of rounds. In deterministic polynomial-time one can feasibly "route in rounds", the number of rounds being the minimum of (i) $O(\chi(\mathcal{T})d^{\delta}\log n + d(y_* + \log n))$ for any fixed $\delta \in (0,1)$, (ii) $O(\eta^{-1}d(y_* + \log n))$, if for all i, $d_i \geq \eta$ and (iii) $O\left(\chi(\mathcal{T})\sqrt{m\,(1 + (\log n)/\chi(\mathcal{T}))}\right)$ [11]. Minimizing the number of rounds for UFP is related to wavelength assignment in optical networks. Connections routed in the same round can be viewed as being assigned the same wavelength. There is a burgeoning literature dealing with *path coloring* as this problem is often called; usually the focus is on special graph classes. See [101, Chapter 2] for an introduction to this area.

The high-capacity case. In the *high-capacity* UFP, the minimum edge capacity is $\Omega(\log m)$ times the maximum demand. An optimal deterministic $O(\log n)$-competitive online algorithm was obtained by Awerbuch, Azar and Plotkin [6]. It maintains length functions for the edges that are exponential in the current load. This idea was introduced for multicommodity flow in [90] and heavily used thereafter (see, e.g., [67, 79, 105, 39]). Raghavan [81] showed that standard randomized rounding achieves with high probability an $O(1)$-approximation for maximum weight with respect to the fractional optimum. Similarly, one obtains that the high-capacity UFP admits an $O(1)$-approximation for congestion. In general, if $d_{max} \leq u_{min}/B$, for some $B > 1$, various improved bounds that depend on B exist, some obtainable via combinatorial algorithms. See [62, 11, 7, 64, 14] for details. Some particularly good results have been obtained for the *half-disjoint* case, i.e., when $B = 2$ [53, 16, 77].

3.2 Packing Integer Programs and UFP. Given $A \in [0,1]^{M \times N}$, $b \in [1,\infty)^M$ and $c \in [0,1]^N$ with $\max_j c_j = 1$, a *packing integer program (PIP)* $\mathcal{P} = (A, b, c)$ seeks to maximize $c^T \cdot x$ subject to $x \in Z_+^N$ and $Ax \leq b$. Constraints of the form $0 \leq x_j \leq d_j$ are clearly allowed. Let B and ζ denote respectively $\min_i b_i$, and the maximum number of non-zero entries in any column of A. The restrictions on the values in

A, b, c are without loss of generality; arbitrary nonnegative values can be scaled appropriately [98]. When $A \in \{0,1\}^{M \times N}$, we say that we have a $(0,1)$-*PIP*. The best guarantees known for PIPs are due to Srinivasan; those for $(0,1)$-PIPs are better than those known for general PIPs by as much as an $\Omega(\sqrt{M})$ factor [98, 96].

As witnessed by the (LP-MCF) relaxation, UFP is a packing problem, albeit one with an exponential number of variables. Motivated by UFP, [62] defined the class of *column-restricted PIPs (CPIPs):* these are the PIPs in which all nonzero entries of column j of A have the same value ρ_j, for all j. Observe that a CPIP generalizes Knapsack. If one obtains the fractional solution f to the (LP-MCF) relaxation, one can formulate the *rounding problem* as a polynomial-size CPIP where the columns of A correspond to the paths used in the fractional solution and the rows correspond to edges in the graph, hence to capacity constraints. The column value ρ_j equals the demand d_j of the commodity corresponding to the path represented by the column. A preprocessing step requires to transform first the fractional solution to a *fractional single-path solution.* This is a fractional solution in which (i) at most one path per commodity is used and (ii) if a commodity is routed at least a $\Omega(1/\log m)$ fraction of the demand is sent to the sink [62]. In combination with improved bounds for CPIPs this approach yielded the $O(\sqrt{m}\log m)$-approximation for UFP mentioned above. The fractional single-path solution concept resurfaced in the algorithm for EDP on DAGs in [15] (cf. Par. 2.3).

A result of independent interest in [62] shows that any family of column-restricted PIPs can be approximated asymptotically as well as the corresponding family of $(0,1)$-PIPs. This result is obtained constructively via the *grouping-and-scaling* technique which first appeared in [61] in the context of single-source UFP (see Par. 3.4 below). Let z_* be the fractional optimum. For a general CPIP the result of [62] translates to the existence of an integral solution of value $\Omega\left(\max\left\{\frac{z_*}{M^{1/(\lfloor B\rfloor+1)}}, \frac{z_*}{\zeta^{1/\lfloor B\rfloor}}, z_*\left(\frac{z_*}{M\log\log M}\right)^{1/\lfloor B\rfloor}\right\}\right)$. Baveja and Srinivasan [10] improved the dilation bound for column-restricted PIPs to $\Omega(\frac{z_*}{t^{1/\lfloor B\rfloor}})$ where $t \leq \zeta$ is the maximum column sum of A.

3.3 Combinatorial algorithms and other results. For extended UFP with polynomially bounded demands, [44] gave a simple randomized algorithm that achieves an $O(\sqrt{m}\log^{3/2} m)$-approximation and generalized the greedy algorithm for EDP [62] (cf. Par. 2.2) to UFP, to obtain an $O(\sqrt{m}\log^2 m)$-approximation. Azar and Regev [7] provided the first strongly-polynomial algorithm for weighted UFP that achieves an

$O(\sqrt{m})$-approximation. For weighted extended UFP they obtained a strongly-polynomial $O(\sqrt{m}\log(2 + \frac{d_{max}}{u_{min}}))$-approximation algorithm. By a reduction from the two-pair decision problem, [7] shows it is NP-hard to obtain an $O(n^{1-\varepsilon})$-approximation for extended weighted UFP, for any fixed $\varepsilon > 0$. The lower bound applies with all the commodities sharing the same source but with weights different from the demands. For extended UFP the integrality gap of (LP-MCF) is $\Omega(n)$ even when the input graph is a path [14].

Further progress in terms of greedy algorithms was achieved by Kolman and Scheideler [64] and Kolman [63]. Recall the BGA algorithm from Par. 2.2. Kolman and Scheideler proposed the *careful BGA,* parameterized by L. The commodities are ordered according to their demands, starting with the largest. Commodity i is accepted if there is a feasible path P for it such that, after routing i, the total flow is larger than half their capacity on at most L edges of P. Let $\mathcal{B}_1$ be the solution thus obtained and $\mathcal{B}_2$ the solution consisting simply of the largest demand routed on any path. The output is $\mathcal{B} := \max\{\mathcal{B}_1, \mathcal{B}_2\}$. In [64] the careful BGA is shown to achieve an $O(\sqrt{m})$-approximation for extended UFP. Generalizing Theorem 1.1 above to UFP, Kolman showed that the careful BGA achieves an $O(\min\{\sqrt{m}, n^{2/3}\})$-approximation on undirected networks and $O(\min\{\sqrt{m}, n^{4/5}\})$-approximation on directed networks, even for extended UFP. Currently these are the best published bounds for UFP; previously they had been shown for UCUFP in [15]. Recently, Chekuri et al. [18] obtained an LP-based $O(\sqrt{n})$-approximation algorithm for UFP on undirected graphs.

Guarantees depending on the network structure. Existing approximation guarantees for UFP are rather weak and on directed graphs one cannot hope for significant improvements, unless P= NP. A different line of work has aimed for approximation ratios depending on parameters other than n and m. This type of work was originally motivated in part by popular hypercube-derived interconnection networks (cf. [85]). Theoretical advances on these networks are typically facilitated by their rich expansion properties. A graph $G = (V, E)$ is an α-expander if for every set X of at most half the vertices, the number of edges leaving X is at least $\alpha|X|$. Concluding a long line of research, Frieze [37] showed that in any r-regular graph with sufficiently strong expansion properties and r a sufficiently large constant, *any* $\Omega(n/\log n)$ vertex pairs can be connected via edge-disjoint paths. See [37] for references on the long history of the topic and the precise underlying assumptions. In such an expander the median distance between pairs of vertices is $O(\log n)$, hence the result of Frieze is within a constant factor of optimal. This basic property, that expanders are rich in short edge-disjoint paths, has been exploited in various guises in the literature. Results for fractional multicommodity flows along short paths were first given by Leighton and Rao [68].

Kleinberg and Rubinfeld analyzed the BGA on expanders in [54]. In the light of Frieze's result above, the BGA achieves an $O(\log n)$-approximation. In [54] it was also shown that for UCUFP one can efficiently compute a fractional solution that routes at least half the maximum demand with dilation $d = O(\Delta^2 \alpha^{-2} \log^3 n)$. Here Δ denotes the maximum degree of the (arbitrary) input graph. The bound on d was improved in [64]. Kolman and Scheideler introduced a new network measure, the *flow number* $F_{G,u}$, and showed that there is always a near optimal fractional flow of dilation $O(F_{G,u})$. The flow number is a quantity computable in polynomial time which is defined based on the solution to a multicommodity flow problem on G. If $u_{min} \geq 1$, $F_{G,u}$ is always $\Omega(\alpha^{-1})$ and $O(\Delta\alpha^{-1} \log n)$ [64]. The BGA examining the demands in nonincreasing order and with $L := 4F_{G,u}$ achieves an $O(F_{G,u})$ approximation for UFP [64]. Chakrabarti et al. [14] provide an $O(F_G \log n)$-approximation for UFP where F_G is a definition of the flow number concept of [64] made independent of capacities. F_G and $F_{G,u}$ coincide on uniform capacity networks. Notably [14] presents an $O(\sqrt{\Delta \log n})$-approximation for UCUFP on Δ-regular graphs with sufficiently strong, in the sense of [37], expansion properties.

3.4 Single-Source Unsplittable Flow. Much better approximation guarantees exist for the case where all commodities share the same source, the so-called *single-source* UFP (SUFP). In contrast to single-source EDP, SUFP is strongly NP-complete [52]. The version of SUFP with costs has also been studied. In the latter problem every edge $e \in E$, has a nonnegative cost c_e. The cost of an unsplittable flow solution is $\sum_{e \in E} c_e f_e$.

The first constant-factor approximations for all the three main objectives (minimizing congestion, maximizing demand and minimizing the number of rounds) were given by Kleinberg [52]. The factors were improved by Kolliopoulos and Stein in [61] where also the first approximations for extended SUFP were given. The grouping-and-scaling technique of [61] consists of partitioning the original problem into a collection of independent subproblems, each of them with demands in a specified range. The fractional solution is then used to assign capacities to each subproblem. The technique is in general useful for translating within constant factors integrality gaps obtained for unit demand instances to arbitrary demand instances. It found further applications, e.g., in approximating CPIPs [62, 10] (cf. Par. 3.2 above), weighted UFP on trees

[19], and [17]. The currently best constant factors for SUFP were obtained by Dinitz, Garg and Goemans [26], though none of them is known to be best possible under some complexity-class separation assumption. Our understanding seems to be better for congestion. The 2-approximation in [26] is best possible if the fractional congestion is used as a lower bound. No ratio better than 3/2 is possible unless $P = NP$. The lower bound comes from minimizing makespan on parallel machines with allocation restriction [69] which reduces in an approximation-preserving manner to SUFP. The mentioned scheduling problem is also a special case of the generalized assignment problem for which a simultaneous $(2, 1)$-approximation for makespan and assignment cost exists [93]. Naturally one wonders whether a simultaneous $(2, 1)$-approximation for congestion and cost is possible for SUFP. This is an outstanding open problem. The currently best tradeoff is a $(3, 1)$-approximation algorithm due to Skutella [94] which cleverly builds on the earlier $(3, 2)$-approximation in [61]. Erlebach and Hall [29] show that it is NP-hard to obtain a $(2 - \varepsilon, 1)$-approximation, for any fixed $\varepsilon > 0$. Experimental evaluations of algorithms for congestion can be found in [60, 59].

4 Variants of the basic problems

In this section we examine some variants of the basic problems. In the *bounded-length* EDP (BLEDP), an additional input parameter M is specified. One seeks a maximum-cardinality set of disjoint $s_i - t_i$ paths, $i = 1, \ldots, k$, under the constraint that the length of each path is at most M. In (s, t)-BLEDP all the pairs share the same source s and sink t. Cases that used to be tractable become NP-hard with the length constraint. Both in the vertex and the edge-disjoint case, (s, t)-BLEDP is NP-complete on undirected graphs even when M is fixed [46]. For variable M and fixed k, the problems remain NP-complete [70]. It is NP-hard to approximate (s, t)-BLEDP within $O(n^{1/2-\varepsilon})$ on directed graphs and, unless $NP = ZPP$, BLEDP cannot be approximated in polynomial time within $O(n^{1/2-\varepsilon})$ on undirected graphs, for any fixed $\varepsilon > 0$ [44]. On the positive side it is easy to obtain an $O(\sqrt{m})$-approximation for BLEDP. For the paths in the optimal solution with length at most $M' := \min\{\sqrt{m}, M\}$ the BGA with parameter $L = M'$ achieves an $O(M')$-approximation. This is because, in the notation of Par. 2.2, $\mathcal{O}_2$ is empty. On the other hand there are at most $\sqrt{m}$ edge-disjoint paths of length more than $\sqrt{m}$. See [44] for other algorithmic results.

In transportation logistics a commodity may be splittable in different containers, each of them to be routed along a single path. One wishes to bound the number of containers used. This motivates the *b-splittable flow problem*, a relaxed version of UFP where a commodity can be split along *at most* $b \geq 1$ paths, b an input parameter. This problem was introduced and first studied by Baier, Köhler and Skutella [9]. Clearly for $b = m$, it reduces to solving the fractional relaxation; it is NP-complete for $b = 2$. See [73, 56] for a continuation of the work in [9]. The author observes in [58] that the single-source 2-splittable flow problem admits a simultaneous $(2, 1)$-approximation for congestion and cost. Finally, a problem in a sense complementary to b-splittable flow and with more history is the *multiroute flow* where for reliability purposes the flow *has to* be split along a given number of edge-disjoint paths. See [8] for definitions and background.

Acknowledgements. Thanks to Chandra Chekuri, Sanjeev Khanna, Maren Martens, Martin Skutella, and Cliff Stein for valuable comments and suggestions. Thanks to Naveen Garg for a clarification on [41], to Aris Pagourtzis for pointing out [101], and to Lex Schrijver for information on EDP on planar graphs.

References

[1] N. Alon and J. Spencer. *The Probabilistic method, 2nd edition.* John Wiley and Sons, 2000.

[2] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of undirected edge disjoint paths with congestion. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 226–244, 2005.

[3] M. Andrews and L. Zhang. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th annual ACM Symposium on Theory of Computing*, pages 284–293, 2005.

[4] M. Andrews and L. Zhang. Hardness of the undirected edge-disjoint paths problem. In *Proceedings of the 37th annual ACM Symposium on Theory of Computing*, pages 276–283, 2005.

[5] Y. Aumann and Y. Rabani. Improved bounds for all-optical routing. In *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, pages 567–576, 1995.

[6] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[7] Y. Azar and O. Regev. Strongly polynomial algorithms for the unsplittable flow problem. In *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization*, pages 15–29, 2001.

[8] A. Bagchi, A. Chaudhary, and P. Kolman. Short length Menger's Theorem and reliable optical routing. *Theoretical Computer Science*, 339:315–332, 2005. Prelim. version in SPAA 03 (revue paper).

[9] G. Baier, E. Köhler, and M. Skutella. On the k-splittable flow problem. *Algorithmica*, 42:231–248, 2005. Special issue on ESA 2002.

[10] A. Baveja and A. Srinivasan. Approximating low-congestion routing and column-restricted packing problems. *Information Processing Letters*, 74:19–25, 2000.

[11] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, 25:255–280, 2000.

[12] D. Bienstock and G. Iyengar. Solving fractional packing problems in $O^*(1/\varepsilon)$ iterations. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 146–155, 2004.

[13] D. Bienstock and M. A. Langston. Algorithmic implications of the Graph Minor Theorem. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Handbook in Operations Research and Management Science 7: Network models*. North-Holland, 1995.

[14] A. Chakrabarti, C. Chekuri, A. Gupta, and A. Kumar. Approximation algorithms for the unsplittable flow problem. In *Proc. APPROX '02*, pages 51–66. Springer-Verlag, 2002.

[15] C. Chekuri and S. Khanna. Edge disjoint paths revisited. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 628–637, 2003.

[16] C. Chekuri, S. Khanna, and B. Shepherd. Edge-disjoint paths in planar graphs. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 71–80, 2004.

[17] C. Chekuri, S. Khanna, and B. Shepherd. Multicommodity flow, well-linked terminals and routing problems. In *Proceedings of the 37th annual ACM Symposium on Theory of Computing*, pages 183–192, 2005.

[18] C. Chekuri, S. Khanna, and B. Shepherd. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, 2:137–146, 2006.

[19] C. Chekuri, M. Mydlarz, and F. B. Shepherd. Multicommodity demand flow in a tree. In *Proceedings of the 30th ICALP*, pages 410–425, 2003.

[20] H. Chernoff. A measure of the asymptotic efficiency for tests of a hypothesis based on sum of observations. *Ann. Math. Stat.*, 23:493–509, 1952.

[21] J. Chuzhoy and S. Naor. New hardness results for congestion minimization and machine scheduling. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 28–34, 2004.

[22] S. Cosares and I. Saniee. An optimization problem related balancing loads on sonet rings. *Telecommunications Systems*, 3:165–181, 1994. Prelim. version as Technical Memorandum. Bellcore, Morristown, NJ, 1992.

[23] M. C. Costa, L. Létocart, and F. Roupin. A greedy algorithm for multicut and integral multiflow in rooted trees. *Operations Research Letters*, 31:21–27, 2003.

[24] M. C. Costa, L. Létocart, and F. Roupin. Minimal multicut and maximum integer multiflow: a survey. *European Journal of Operational Research*, 162:55–69, 2005.

[25] E. A. Dinitz. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Soviet Math. Dokl.*, 11:1277–1280, 1970.

[26] Y. Dinitz, N. Garg, and M. X. Goemans. On the single-source unsplittable flow problem. *Combinatorica*, 19:1–25, 1999. Preliminary version in FOCS 98.

[27] P. Elias, A. Feinstein, and C. E. Shannon. Note on maximum flow through a network. *IRE Transactions on Information Theory IT-2*, pages 117–199, 1956.

[28] P. Erdős and J. L. Selfridge. On a combinatorial game. *Journal of Combinatorial Theory A*, 14:298–301, 1973.

[29] T. Erlebach and A. Hall. NP-hardness of broadcast sheduling and inapproximability of single-source unsplittable min-cost flow. *Journal of Scheduling*, 7:223–241, 2004. Prelim. version in SODA 02.

[30] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5:691–703, 1976.

[31] S. Even and R. E. Tarjan. Network flow and testing graph connectivity. *SIAM Journal on Computing*, 4:507–518, 1975.

[32] L. Fleischer. Approximating fractional multicommodity flows independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 13:505–520, 2000. Prelim. version in FOCS 99.

[33] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canad. J. Math.*, 8:399–404, 1956.

[34] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10:111–121, 1980.

[35] A. Frank. Packing paths, cuts and circuits - a survey. In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, editors, *Paths, Flows and VLSI-Layout*, pages 49–100. Springer-Verlag, Berlin, 1990.

[36] A. Frank. Connectivity and network flows. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 111–177. North-Holland, 1995.

[37] A. M. Frieze. Edge-disjoint paths in expander graphs. *SIAM Journal on Computing*, 30:1790–1801, 2001. Prelim. version in SODA 00.

[38] D. R. Fulkerson and G. B. Dantzig. Computation of maximum flow in networks. *Naval Research Logistics Quarterly*, 2:277–283, 1955.

[39] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[40] N. Garg, V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25:235–251, 1996. Prelim. version in STOC 93.

[41] N. Garg, V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18:3–20, 1997. Prelim. version in ICALP 93.

[42] A. Goldberg. A natural randomization strategy for multicommodity flow and related algorithms. *Information Processing Letters*, 42:249–256, 1992.

[43] M. D. Grigoriadis and L. G. Khachiyan. Fast approximation schemes for convex programs with many blocks and coupling constraints. *SIAM Journal on Optimization*, 4(1):86–107, February 1994.

[44] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *Journal of Computer and System Sciences*, 67:473–496, 2003. Prelim. version in STOC 99.

[45] T. C. Hu. Multi-commodity network flows. *Operations Research*, 11:344–360, 1963.

[46] A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12:277–286, 1982.

[47] G. Karakostas. Faster approximation schemes for fractional multicommodity flow problems. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 166–173, 2002.

[48] D. R. Karger and M. S. Levine. Finding maximum flows in simple undirected graphs seems easier than bipartite matching. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, 1998.

[49] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[50] P. Klein, S. A. Plotkin, C. Stein, and É. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM Journal on Computing*, 23:466–487, 1994. Prelim. version in STOC 90.

[51] J. M. Kleinberg. *Approximation algorithms for disjoint paths problems.* PhD thesis, MIT, Cambridge, MA, May 1996.

[52] J. M. Kleinberg. Single-source unsplittable flow. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–77, October 1996.

[53] J. M. Kleinberg. Decision algorithms for unsplittable flow and the half-disjoint paths problem. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 530–539, 1998.

[54] J. M. Kleinberg and R. Rubinfeld. Short paths in expander graphs. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 86–95, 1996.

[55] J. M. Kleinberg and É. Tardos. Disjoint paths in densely-embedded graphs. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 52–61, 1995.

[56] R. Koch, M. Skutella, and I. Spenke. Approximation and complexity of k-splittable flows. In *3rd Int. Workshop on Approximation and Online Algorithms (WAOA 05)*, volume 3879 of *Lecture Notes in Computer Science*, pages 244–257. Springer-Verlag, 2006.

[57] S. G. Kolliopoulos. *Exact and Approximation Algorithms for Network Flow and Disjoint-Path Problems.* PhD thesis, Dartmouth College, Hanover, NH, August 1998.

[58] S. G. Kolliopoulos. Minimum-cost single-source 2-splittable flow. *Information Processing Letters*, 94:15–18, 2005.

[59] S. G. Kolliopoulos and J. Du. Implementing approximation algorithms for single-source unsplittable flow. In C. C. Ribeiro and S. L. Martins, editors, *Proc. 3rd Int. Workshop on Efficient and Experimental Algorithms*, volume 3059 of *Lecture Notes in Computer Science*, pages 213–227. Springer-Verlag, 2004.

[60] S. G. Kolliopoulos and C. Stein. Experimental evaluation of approximation algorithms for single-source unsplittable flow. In G. Cornuéjols, R. E. Burkard, and G. J. Woeginger, editors, *Proceedings of the 7th Conference on Integer Programming and Combinatorial Optimization*, volume 1610 of *Lecture Notes in Computer Science*, pages 328–344. Springer-Verlag, June 1999.

[61] S. G. Kolliopoulos and C. Stein. Approximation algorithms for single-source unsplittable flow. *SIAM Journal on Computing*, 31:919–946, 2002. Prelim. version in FOCS 97.

[62] S. G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using packing integer programs. *Mathematical Programming A*, 99:63–87, 2004. Prelim. version, titled "Approximating disjoint-path problems using greedy algorithms and packing integer programs" in IPCO 98.

[63] P. Kolman. A note on the greedy algorithm for the unsplittable flow problem. *Information Processing Letters*, 88:101–105, 2003.

[64] P. Kolman and C. Scheideler. Improved bounds for the unsplittable flow problem. *Journal of Algorithms*, 61:20–44, 2006. Prelim. version in SODA 02.

[65] M. R. Kramer and J. van Leeuwen. The complexity of wire-routing and finding minimum-area layouts for arbitrary VLSI circuits. In F. P. Preparata, editor, *VLSI Theory*, volume 2 of *Advances in Computing Research*, pages 129–146. JAI Press, Greenwich, CT, 1984.

[66] F. T. Leighton, S. Rao, and A. Srinivasan. Multicommodity flow and circuit switching. In *Hawaii International Conference on System Sciences*, pages 459–465, 1998.

[67] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problems. *Journal of Computer and System Sciences*, 50:228–243, 1995. Prelim. version in STOC 91.

[68] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999. Prelim. version in FOCS 88.

[69] J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming A*, 46:259–271, 1990.

[70] C. Li, T. McCormick, and D. Simchi-Levi. The complexity of finding two disjoint paths with min-max objective function. *Discrete Applied Mathematics*, 26:105–115, 1990.

[71] J. F. Lynch. The equivalence of theorem proving and the interconnection problem. *ACM SIGDA Newsletter*, 5:31–36, 1975.

[72] B. Ma and L. Wang. On the inapproximability of disjoint paths and minimum Steiner forest with bandwidth constraints. *Journal of Computer and System Sciences*, 60:1–12, 2000.

[73] M. Martens and M. Skutella. Flows on few paths: algorithms and lower bounds. In S. Albers and T. Radzik, editors, *Proceedings of the 12th Annual European Symposium on Algorithms*, volume 3221 of *Lecture Notes in Computer Science*, pages 520–531. Springer-Verlag, 2004.

[74] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[75] R. H. Möhring and D. Wagner. Combinatorial topics in VLSI design, annotated bibliography. In Mauro Dell'Amico, Francesco Maffioli, and Silvano Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 429–444. Wiley, 1997.

[76] R. H. Möhring, D. Wagner, and F. Wagner. VLSI network design: a survey. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Handbooks in Operations Research/Management Science, Volume on Networks*, pages 625–712. North-Holland, 1995.

[77] T. Nguyen. On the disjoint paths problem. *Operations Research Letters*, 35:10–16, 2007.

[78] Y. Perl and Y. Shiloach. Finding two disjoint paths between two pairs of vertices in a graph. *Journal of the ACM*, 25:1–9, 1978.

[79] S. Plotkin, D. B. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20:257–301, 1995.

[80] T. Radzik. Fast deterministic approximation for the multicommodity flow problem. In *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, pages 486–496, 1995.

[81] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *Journal of Computer and System Sciences*, 37:130–143, 1988.

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

[83] H. Ripphausen-Lipa, D. Wagner, and K. Weihe. Survey on efficient algorithms for disjoint paths problems in planar graphs. In W. Cook, L. Lovász, and P. D. Seymour, editors, *DIMACS-Series in Discrete Mathematics and Theoretical Computer Science, Volume 20 on the "Year of Combinatorial Optimization"*, pages 295–354. AMS, 1995.

[84] N. Robertson and P. D. Seymour. Graph Minors XIII. The disjoint paths problem. *Journal of Combinatorial Theory B*, 63:65–110, 1995.

[85] C. Scheideler. *Universal routing strategies for interconnection networks*, volume 1390 of *LNCS*. Springer-Verlag, Berlin, 1998.

[86] A. Schrijver. Homotopic routing methods. In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, editors, *Paths, Flows and VLSI-Layout*. Springer-Verlag, Berlin, 1990.

[87] A. Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM Journal on Computing*, 23:780–788, 1994.

[88] A. Schrijver. *Combinatorial Optimization: polyhedra and efficiency*. Springer-Verlag, Berlin, 2003.

[89] P. D. Seymour. Disjoint paths in graphs. *Discrete Mathematics*, 29:293–309, 1980.

[90] F. Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *Journal of the ACM*, 37:318 – 334, 1990.

[91] Y. Shiloach. A polynomial solution to the undirected two paths problem. *Journal of the ACM*, 27:445–456, 1980.

[92] D. B. Shmoys. Cut problems and their applications to Divide and Conquer. In D. S. Hochbaum, editor, *Approximation algorithms for NP-hard problems*, pages 192 –231. PWS, Boston, 1997.

[93] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming A*, 62:461–474, 1993.

[94] M. Skutella. Approximating the single-source unsplittable min-cost flow problem. *Mathematical Programming B*, 91:493–514, 2002. Prelim. version in FOCS 00.

[95] J. Spencer. *Ten Lectures on the Probabilistic Method*. SIAM, Philadelphia, 1987.

[96] A. Srinivasan. An extension of the Lovász Local Lemma and its applications to integer programming. In *Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 6–15, 1996.

[97] A. Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow and related routing problems. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 416–425, 1997.

[98] A. Srinivasan. Improved approximations guarantees for packing and covering integer programs. *SIAM Journal on Computing*, 29:648–670, 1999. Prelim. version in STOC 95.

[99] A. Srinivasan. A survey of the role of multicommodity flow and randomization in network design and routing. In P. M. Pardalos, S. Rajasekaran, and J. Rolim, editors, *American Mathematical Society, Series in Discrete Mathematics and Theoretical Computer Science*, volume 43, pages 271–302. 1999.

[100] A. Srinivasan. New approaches to covering and packing problems. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 567–576, 2001.

[101] S. Stefanakos. *On the Design and Operation of High-Performance Optical Networks*. PhD thesis, ETH Zurich, No. 15691, 2004.

[102] C. Thomassen. 2-linked graphs. *European Journal of Combinatorics*, 1:371–378, 1980.

[103] K. Varadarajan and G. Venkataraman. Graph decomposition and a greedy algorithm for edge-disjoint paths. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, pages 379–380, 2004.

[104] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, 2001.

[105] N. E. Young. Randomized rounding without solving the linear program. In *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, pages 170–178, 1995.

Index