The Maximum Integer Multiterminal Flow Problem

Cédric Bentz

CEDRIC-CNAM, 292, Rue Saint-Martin,
75141 Paris Cedex 03, France
cedric.bentz@cnam.fr

Abstract. Given an edge-capacitated graph and k *terminal* vertices, the *maximum integer multiterminal flow* problem (MaxIMTF) is to route the maximum number of flow units between the terminals. For directed graphs, we introduce a new parameter $k_L \leq k$ and prove that MaxIMTF is $\mathcal{NP}$-hard when $k = k_L = 2$ and when $k_L = 1$ and $k = 3$, and polynomial-time solvable when $k_L = 0$ and when $k_L = 1$ and $k = 2$. We also give an $2\log_2(k_L + 2)$-approximation algorithm for the general case. For undirected graphs, we give a family of valid inequalities for MaxIMTF that has several interesting consequences, and show a correspondence with valid inequalities known for MaxIMTF and for the associated *minimum multiterminal cut problem*.

1 Introduction

Routing problems in networks are commonly modeled by flow or multicommodity flow problems. Given an edge-capacitated graph (directed or undirected), the goal is to route flow units (requests) between prespecified vertices. When one seeks to route the maximum number of flow units from a unique source to a unique sink, the problem is the famous *maximum flow problem.* The Ford-Fulkerson's theorem [11] gives a good characterization for this case, which is efficiently solvable [1]. In particular, this theorem states that, if the capacities are integral, the value of a maximum integer flow is equal to the value of a minimum cut, i.e., to the value of a minimum weight set of edges whose removal separates the source from the sink. Unfortunately, this does not hold for more general variants. One of the most studied variant is the *maximum integer multicommodity flow problem*: given an edge-capacitated graph $G = (V, E)$ and a list of source-sink pairs, the goal is to simultaneously route the maximum number of flow units, each unit being routed from one source to its corresponding sink.

This problem is $\mathcal{NP}$-hard even for two source-sink pairs [10], and cannot be approximated within $|E|^{1/2-\epsilon}$ (resp. within $(\log |E|)^{1/3-\epsilon}$) for every $\epsilon > 0$ in directed graphs [15] (resp. in undirected graphs [2]) unless $\mathcal{P} = \mathcal{NP}$ (recall that, for a maximization (resp. minimization) problem, an α-approximation algorithm is a polynomial-time algorithm that always outputs a feasible solution whose value is at least $1/\alpha$ times (resp. at most α times) the value of an optimal solution). The corresponding generalization of the problem of finding a minimum cut is the

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3982, pp. 738–747, 2006.

minimum multicut problem, which asks to select a minimum weight set of edges whose removal separates each source from its corresponding sink. This problem is also $\mathcal{NP}$-hard in several special cases, and has a noticeable relationship with the former: the continuous relaxations of the linear programming formulations of the two problems are dual [8]. In particular, this interesting property has been used to design good approximation algorithms for both problems [14]. Further results and references concerning these problems can be found in [1] and [8].

Another generalization of the maximum flow problem is the *maximum integer multiterminal flow problem* (MaxIMTF): given an edge-capacitated graph and a set $T = \{t_1, \ldots, t_k\}$ of *terminal* vertices, MaxIMTF is to route the maximum number of flow units between the terminals. Note that this problem is a particular maximum integer multicommodity flow problem in which the source-sink pairs are (t_i, t_j) for $i \neq j$. The associated *minimum multiterminal cut problem* (MinMTC) is to select a minimum weight set of edges whose removal separates t_i from t_j for $i \neq j$. Note that MaxIMTF and MinMTC also have the duality relationship mentioned above. MinMTC has been widely studied in the undirected case (see [3], [5], [6], [7], [8], [9], [16] and [20]), and the directed case has also received some attention: Garg et al. [13] show that it is $\mathcal{NP}$-hard even for $k = 2$ and give an $2\log_2 k$-approximation algorithm, and Naor and Zosin [18] give a 2-approximation algorithm. However, the algorithm of Garg et al. has an interesting property: it computes a multiterminal cut whose value is at most $2\log_2 k$ times the value of an integer multiterminal flow, and hence is an $2\log_2 k$-approximation for both MinMTC *and* MaxIMTF (while the algorithm of Naor and Zosin does not provide an approximate solution for MaxIMTF). (Note that the same idea easily yields an $\log_2 k$-approximation algorithm for MaxIMTF in undirected graphs.) Costa et al. [8] show that MaxIMTF and MinMTC are polynomial-time solvable in acyclic directed graphs by using a simple reduction to a maximum flow and a minimum cut problem, respectively. To the best of our knowledge, these are the only results about MaxIMTF in directed graphs. In undirected graphs, MaxIMTF has recently be shown to be polynomial-time solvable by using the ellipsoid method [17] (the result is based on the associated Mader's theorem on T-paths [19, Chap. 73]). Algorithmic aspects of special cases have also been studied (inner eulerian graphs in [12] and trees in [4]). However, it can be easily noticed that, for all the problems mentioned above, the general directed case is "harder" than the undirected one, since there exists a linear reduction from the latter to the former: simply replace each edge by the gadget given in [19, (70.9) on p. 1224].

The motivation of this paper is to explore further the complexity of MaxIMTF. Given a directed graph, we say that a terminal is *lonely* if it lies on at least one directed cycle containing no other terminal, and we let T_L denote the set of lonely terminals and $k_L = |T_L|$. We shall see that k_L is a key parameter for better understanding the complexity and approximability of MaxIMTF. Moreover, some of our results will extend to MinMTC.

We first show that MaxIMTF is strongly $\mathcal{NP}$-hard in directed graphs, even if $k_L = k = 2$ or if $k_L = 1$ and $k = 3$ (Section 2). Then, we prove MaxIMTF to be tractable when $k_L = 0$ and when $k_L = 1$ and $k = 2$, and improve the $2\log_2 k$-approximation algorithm of Garg et al. [13] by providing an $2\log_2(k_L + 2)$-approximation algorithm for the general case (Section 3). Eventually, we give a family of valid inequalities for MaxIMTF in undirected graphs, and show an interesting correspondence with valid inequalities known for the associated problem MinMTC (Section 4).

Note that, throughout this paper, we consider only *simple* graphs. We call *Directed* (resp. *Undirected*) MaxIMTF the problem MaxIMTF defined in directed (resp. undirected) graphs. Moreover, due to lack of space, we sometimes omit some details in our proofs.

2 $\mathcal{NP}$-Hardness Proof

We show in this section that Directed MaxIMTF is strongly $\mathcal{NP}$-hard, even if $k = k_L = 2$ (or $k_L = 1$ and $k = 3$). In order to do this, we adapt the proof, given in [10], of the $\mathcal{NP}$-completeness of the *directed integer multicommodity flow problem* with two source-sink pairs, (s_1, s_1') and (s_2, s_2'): given an arc-capacitated directed graph $G = (V, A)$ and two integer demands d_1 and d_2 associated with the respective source-sink pairs, it asks to decide whether these demands can be simultaneously routed while respecting the capacity constraints (if, for an instance, the answer is *yes*, then this instance is *solvable*). In the instance used in the proof of [10, Theorem 3], $d_1 = 1$, $d_2 \le |V|$ and all the arcs have capacity 1. Moreover, this instance satisfies

$$|\Gamma^-(s_1)| = |\Gamma^-(s_2)| = 0$$

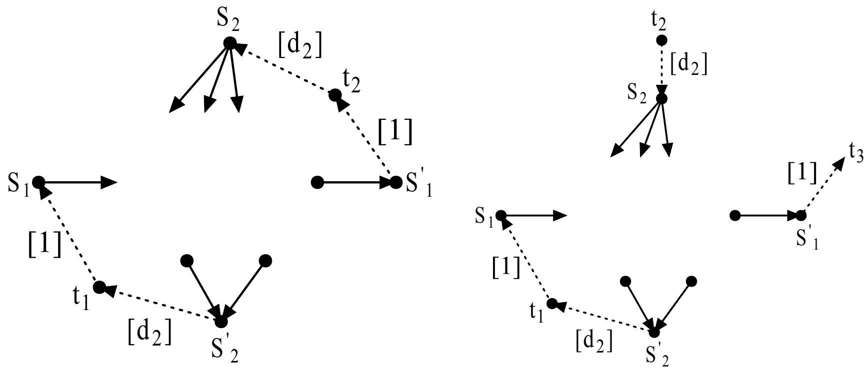
and

$$|\Gamma^+(s_1')| = |\Gamma^+(s_2')| = 0$$

where, for $v \in V$, $\Gamma^+(v) = \{u \in V \text{ such that } (v,u) \in A\}$ and $\Gamma^-(v) = \{u \in V \text{ such that } (u,v) \in A\}$. We modify this initial instance as follows: we add two new vertices, t_1 and t_2, and four arcs (t_1, s_1), (s_2', t_1), (t_2, s_2) and (s_1', t_2), valued by 1, d_2, d_2 and 1 respectively (see Fig. 1(a)).

It is easy to see that the initial instance is solvable if and only if the optimum value for the maximum integer multicommodity flow instance defined on the pairs (t_1, t_2) and (t_2, t_1) is equal to $d_2 + 1$ (no flow unit being routed from s_i to s_j, from s_i' to s_j' for $i \neq j$, or from s_i to s_j' for $i \neq j$). Moreover, the latter instance is equivalent to a directed maximum integer multiterminal flow instance with two terminals, t_1 and t_2. Eventually, we can replace each one of the two arcs (s_2', t_1) and (t_2, s_2) by d_2 directed paths of length two (containing only arcs with capacity 1) between the corresponding endpoints, and obtain an instance of MaxIMTF where each arc has capacity 1. The described transformation is clearly polynomial, and hence

Theorem 1. Directed MaxIMTF *is $\mathcal{NP}$-hard in graphs with unit capacities, even with only two terminals.*



(a) Reduction for $k_L = 2$ and $k = 2$ (b) Reduction for $k_L = 1$ and $k = 3$

Fig. 1. Reductions for Directed MaxIMTF

In particular, this implies the strong $\mathcal{NP}$-hardness of Directed MaxIMTF. It can also be noticed that this result matches the complexity result for the associated cut problem MinMTC in directed graphs [13].

However, in the proof of Theorem 1, $k = k_L = 2$: so, what happens when $k_L = 1$? Actually, a slightly different proof shows that Directed MaxIMTF remains $\mathcal{NP}$-hard, even with unit capacities. We define three new terminals instead of two: t_1, t_2 and t_3. Moreover, we add four arcs (t_1, s_1), (s'_2, t_1), (t_2, s_2) and (s'_1, t_3), valued by 1, d_2, d_2 and 1 respectively (note that $T_L = \{t_1\}$; see Fig. 1(b)). In this instance, it is easy to see that there exists an integer multiterminal flow of value $d_2 + 1$ if and only if 1 flow unit is routed from t_1 to t_3 and d_2 flow units are routed from t_2 to t_1. This implies:

Theorem 2. Directed MaxIMTF *is $\mathcal{NP}$-hard in graphs with unit capacities, even if $k_L = 1$ and $k = 3$.*

We shall deal with the cases where $k_L = 0$ and where $k_L = 1$ and $k = 2$ in the next section.

3 Exact and Approximation Algorithms

From the previous section, Directed MaxIMTF is strongly $\mathcal{NP}$-hard even for $k_L = 1$ and $k = 3$ and for $k = k_L = 2$. Hence, if $\mathcal{P} \neq \mathcal{NP}$, the only efficient algorithms one can expect to design are approximation algorithms. In this section, we improve the $2\log_2 k$-approximation algorithm of Garg et al. [13] and give an $2\log_2(k_L + 2)$-approximation algorithm for Directed MaxIMTF.

The basic idea of our approach is to combine the algorithm of Garg et al. with an interesting strengthening of [8, Proposition 3]. The main idea of the proof of [8, Proposition 3] (that shows that MaxIMTF and MinMTC are polynomial-time solvable in acyclic directed graphs) is to split up each terminal vertex t_i

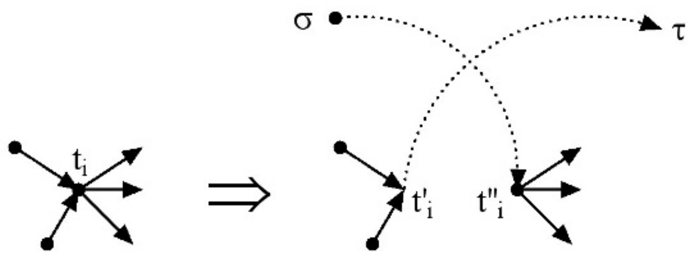


Fig. 2. Splitting up terminal t_i

into two new vertices, t'_i and t''_i, such that all the vertices in $\Gamma^-(t_i)$ are linked to t'_i and t''_i is linked only to the vertices in $\Gamma^+(t_i)$. Then, we add two new vertices, σ and τ, and link (by arcs with sufficiently large capacities) every t'_i to τ and σ to every t''_i (see Fig. 2). Finally, we compute a maximum flow between σ and τ (obviously, we assume that the capacities are integral). The obtained flow is a valid integer multiterminal flow for the initial instance if, in the modified instance, no flow unit is routed from t''_i to t'_i for some i.

The main point for us is that, if there is no lonely terminal, then, by splitting up the terminals as explained, there will remain no directed path from t''_i to t'_i for each i, and hence we will be able to solve MaxIMTF and MinMTC using the above technique. Actually, if we want to guarantee that, after splitting up each terminal, the modified graph does not admit a directed path from t''_i to t'_i for some i (otherwise, we cannot be sure that the flow we will compute in the modified graph will be a valid multiterminal flow in the initial graph), this is essentially the best (i.e., weakest) assumption that can be made. Namely, one can easily show:

Theorem 3. *After splitting up all the terminals, there is no directed path between t''_i and t'_i for each i if and only if $k_L = 0$.*

This also implies the following strengthening of [8, Proposition 3]:

Theorem 4. MinMTC *and* MaxIMTF *are polynomial-time solvable in directed graphs if $k_L = 0$, by using a max flow-min cut algorithm.*

Actually, the last remaining case, i.e., the case where $k_L = 1$ and $k = 2$, is also polynomial-time solvable. Indeed, one can prove that on any directed cycle containing only the terminal in T_L, there is a *removable* arc, i.e., an arc lying on no elementary path between the two terminals (otherwise, there would be a vertex of this cycle lying on another directed cycle containing only the terminal not in T_L, which is impossible). By iteratively removing such arcs, we are back to the case where $k_L = 0$. Hence:

Theorem 5. Directed MaxIMTF *is tractable if $k_L = 1$ and $k = 2$.*

Theorems 3 and 4 show the importance of the parameter k_L for both MaxIMTF and MinMTC. Moreover, this suggests the following approach for finding approximate solutions for these two problems: first, (a) split up each terminal $t_i \in T - T_L$ into t_i' and t_i'' as explained above, add the two vertices σ and τ, and link (by *heavy* arcs) every t_i' to τ and σ to every t_i''; then, (b) compute a multiterminal cut and flow for this new instance (i.e., where the terminal set is $T_L \bigcup \{\sigma, \tau\}$) by using the algorithm of Garg et al. [13]. The definition of T_L guarantees that we obtain a valid integer multiterminal flow.

Hence, the main difference with their algorithm is that, before using their divide-and-conquer strategy, we transform the graph by replacing the terminals in $T - T_L$ by two new terminals, σ and τ. This implies that we use Garg et al.'s algorithm on an instance with $k_L + 2$ terminals, and so we obtain an approximation factor of $2\log_2(k_L + 2)$ (instead of $2\log_2 k$).

Actually, one can even prove that this analysis of the approximation ratio is tight by using a particular family of instances built on an undirected tree with $k = 2^p$ vertices (all vertices are terminal), and which is transformed into a directed graph by replacing each edge by the gadget given in [19, (70.9) on p. 1224] (each arc of the gadget has the capacity of the initial edge). Due to lack of space, we do not give the whole construction here.

4 Polyhedral Results for the Undirected Case

In this section, we give a family of valid inequalities for the LP formulation of Undirected MaxIMTF given in [13]. We call them *tree inequalities*, as they can be seen as the "flow counterpart" of the tree inequalities given in [7] and characterizing completely the polytope of MinMTC in trees (as shown in [7]).

Theorem 6. *Let U be an undirected tree and $X_U = \{t_1, \ldots, t_h\}$ be the terminals in U. Assume the leaves of U coincide with these h terminals (assume without loss of generality that we have removed from U the edges such that no flow unit can be routed through them). Then, if F_U denotes the total number of flow units that are routed between the terminals in X_U, the inequality*

$$F_U \leq \left\lfloor \frac{\sum_{i\in\{1,\ldots h\}} c_i}{2} \right\rfloor$$

is a valid inequality for MaxIMTF *(called* tree inequality*), where, for each i, c_i is the minimum capacity of the edges contained in the path p_i linking t_i to n_i, its nearest vertex of degree at least* 3 *in U (see* Fig. 3*).*

A proof of this theorem is given below. In particular, this proof will imply that these inequalities (together with the usual constraints) are sufficient to guarantee the existence of an integer optimal solution to the continuous relaxation of the linear program (i.e., of an optimal solution for MaxIMTF) in undirected trees (recall that the tree inequalities for MinMTC do give a complete characterization of the associated polytope in undirected trees [7]). These inequalities may also be sufficient to completely characterize the polytope associated with MaxIMTF in undirected trees. On the other hand, a quadratic algorithm for

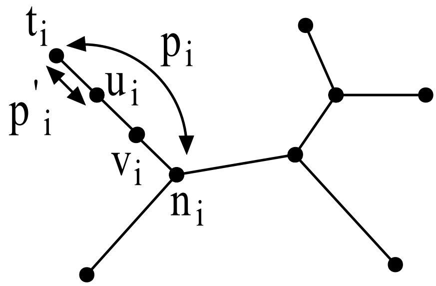


Fig. 3. A tree instance for Theorem 6 ((u_i, v_i) has capacity c_i)

MaxIMTF in trees is already known [4]. Our original motivation came from the fact that the authors of [7] used the description of the polytope associated with MinMTC to derive an efficient algorithm for MinMTC in trees by using complementary slackness conditions: can it be done for MaxIMTF?

Actually, one can prove that the tree inequalities for MaxIMTF are a special case of a more general class of valid inequalities, that we call *inner odd set inequalities*: they have been used very recently (the paper [17] has appeared just after the submission of the present paper) to prove that Undirected MaxIMTF was polynomial-time solvable via the ellipsoid method. They are derived from the fact that evenness considerations are well-known to be of great importance in integer flow problems with several sources and sinks (see [12] for example). They can be defined as follows:

Definition 1. *Let $G = (V, E)$ be an undirected graph and let T be the set of terminal vertices. For each $X \subseteq V \setminus T$, let $(X, V \setminus X)$ be the set of edges lying between X and $V \setminus X$ and let $c(X, V \setminus X)$ be the total capacity of the edges in $(X, V \setminus X)$. Then, F, the total number of flow units routed between the terminals in T, satisfies the following valid inequality (called* inner odd set inequality*)*

$$F \leq \left\lfloor \frac{c(X, V \setminus X)}{2} \right\rfloor .$$

Roughly speaking, the validity of these inequalities comes from the fact that each flow unit in G is routed through no or an even number of edges in $(X, V \setminus X)$, since X contains no terminal (see Fig. 4). Hence, each flow unit routed through an edge in $(X, V \setminus X)$ is counted at least twice, and the total amount of flow in $(X, V \setminus X)$ is thus equal to (at least) $2F$. This amount being at most $c(X, V \setminus X)$, we can divide both sides of the inequality by 2 and use the integrality of F to obtain the desired result.

Now we can prove Theorem 6 by using Definition 1 and the following fact.

Theorem 7. *The tree inequalities are a special case of the inner odd set inequalities.*

Proof. We use the notations of Theorem 6. Given an undirected tree U, for each i, let (u_i, v_i) be an edge of p_i with capacity c_i (u_i lying in the path from t_i to v_i),

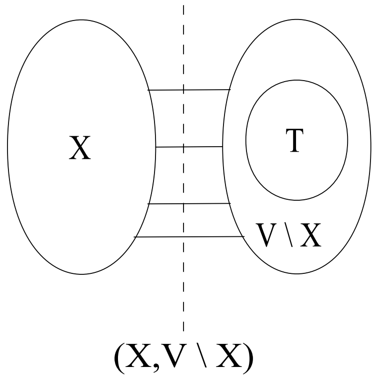


Fig. 4. An inner odd set configuration ($c(X, V \setminus X)$ is odd)

and let p'_i be the path from t_i to u_i (see Fig. 3). Then, the tree inequality on U is obtained by taking the inner odd set inequality defined on the set $X = V \setminus \bigcup_i p'_i$, since any flow unit has to be routed through at least one edge (and, in fact, through exactly two edges) in $(X, V \setminus X)$, i.e., through (u_i, v_i) and (u_j, v_j) for some i and j. □

In trees, it is not difficult to see that all the inner odd set inequalities that matters are the ones corresponding to tree inequalities. Hence, from [17], these inequalities suffice to guarantee the existence of integer optimal solutions in undirected trees.

In fact, there exists an interesting relationship between the inner odd set inequalities and the inner eulerian assumption made in [12]. Given an edge-capacitated undirected graph $G = (V, E)$, the *degree* of a vertex $v \in V$, denoted by $d_V(v)$, is the sum of the capacities of the edges adjacent to v. Moreover, for $X \subseteq V$, $d_X(v)$ is the degree of vertex v in the subgraph of G induced by $X \bigcup \{v\}$. A graph is *inner eulerian* if every non terminal vertex has an even degree. Theorem 8 shows that the inner odd set inequalities are useless if the graph is inner eulerian.

Theorem 8. *Given a graph $G = (V, E)$ and a set of terminal vertices T, G is inner eulerian if and only if $\forall X \subseteq V \setminus T$, $c(X, V \setminus X)$ is even.*

Proof. It is easily seen that, by definition, G is inner eulerian if and only if $\forall X \subseteq V \setminus T$ with $|X| = 1$, $c(X, V \setminus X)$ is even. Now, assume that G is inner eulerian, and let $X \subset V$ contain no terminal. Then, we have

$$c(X, V \setminus X) = \sum_{v \in X} d_{V \setminus X}(v) = \sum_{v \in X} d_V(v) - \sum_{v \in X} d_X(v) \ .$$

Moreover, $\sum_{v \in X} d_V(v)$ is even since G is inner eulerian, and $\sum_{v \in X} d_X(v)$ is always even (because it is equal to two times the sum of the capacities of the edges having both endpoints in X). This implies that $c(X, V \setminus X)$ is even, and Theorem 8 follows. □

An interesting question would be to determine whether the inner odd set inequalities (together with the usual constraints) give a complete characterization of the polytope of Undirected MaxIMTF. Theorem 8 shows that a positive answer to this question would imply that the polytope of the continuous relaxation of the LP formulation of Undirected MaxIMTF is integral in inner eulerian undirected graphs (the existence of integer optimal solutions was already known [12]).

5 Conclusion and Open Problems

The parameter k_L introduced in this paper improves our knowledge of the boundary between tractable and intractable cases of MaxIMTF in directed graphs. Several results, positive and negative, about tractability and approximability of this problem, are provided. Moreover, we have given a family of valid inequalities for the undirected case, and have proved an interesting correspondence with valid inequalities already known. However, two important questions remain open: is there an $O(1)$-approximation algorithm for the general directed case? Moreover, are they other tractable special cases (e.g., planar digraphs)?

References

1. Ahuja, A.K., Magnanti, T.L., Orlin, J.B.: Network Flows – Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs, New Jersey (1993).
2. Andrews, M., Zhang, L.: Hardness of the undirected edge-disjoint paths problem. Proceedings STOC'05 (2005) 276–283.
3. Bertsimas, D., Teo, C.-P., Vohra, R.: Analysis of LP relaxations for multiway and multicut problems. Networks **34** (1999) 102-114.
4. Billionnet, A., Costa, M.-C.: Multiway cut and integer flow problems in trees. In: Liberti, L., Maffioli, F. (eds.): CTW04 Workshop on Graphs and Combinatorial Optimization. Electronic Notes in Discrete Mathematics **17** (2004) 105–109.
5. Călinescu, G., Karloff, H., Rabani, Y.: An improved approximation algorithm for Multiway Cut. Proceedings STOC'98 (1998) 48–52.
6. Chen, D.Z., Wu, X.: Efficient algorithms for k-terminal cuts on planar graphs. Algorithmica **38** (2004) 299–316.
7. Chopra, S., Rao, M.R.: On the multiway cut polyhedron. Networks **21** (1991) 51–89.
8. Costa, M.-C., Létocart, L., Roupin, F.: Minimal multicut and maximal integer multiflow: a survey. European Journal of Operational Research **162** (2005) 55–69. Elsevier.
9. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM Journal On Computing **23** (1994) 864–894.
10. Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM Journal on Computing **5** (1976) 691–703.

11. Ford, L.R., Fulkerson, D.R.: Maximal Flow Through a Network. Canadian Journal of Mathematics **8** (1956) 339–404.
12. Frank, A., Karzanov, A., Sebö, A.: On integer multiflow maximization, SIAM J. Discrete Mathematics **10** (1997) 158–170.
13. Garg, N., Vazirani, V.V., Yannakakis, M.: Multiway cuts in directed and node weighted graphs. In: Abiteboul, S., Shamir, E. (eds.): 21st International Colloquium on Automata, Languages and Programming. Lecture Notes in Computer Science, Vol. 820. Springer-Verlag (1994) 487–498.
14. Garg, N., Vazirani, V.V., Yannakakis, M.: Primal-dual approximation algorithms for integral flow and multicut in trees. Algorithmica **18** (1997) 3–20.
15. Guruswami, V., Khanna, S., Rajaraman, R., Shepherd, B., Yannakakis, M.: Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. Proceedings STOC'99 (1999) 19–28.
16. Karger, D., Klein, P., Stein, C., Thorup, M., Young, N.: Rounding Algorithms for a Geometric Embedding of Minimum Multiway Cut. Proceedings STOC'99 (1999) 668–678.
17. Keijsper, J.C.M., Pendavingh, R.A., Stougie, L.: A linear programming formulation of Mader's edge-disjoint paths problem. Journal of Combinatorial Theory, Series B **96** (2006) 159–163.
18. Naor, J., Zosin, L.: A 2-approximation algorithm for the directed multiway cut problem. Proceedings FOCS'97 (1997) 548–553.
19. Schrijver, A.: Combinatorial Optimization - Polyhedra and Efficiency. Algorithms and Combinatorics **24**. Springer-Verlag (2003).
20. Yeh, W.-C.: A Simple Algorithm for the Planar Multiway Cut Problem. Journal of Algorithms **39** (2001) 68–77.