Multilateral Transport Games

F.B. Shepherd, G.T. Wilfong
Bell Laboratories
600 Mountain Ave.
Murray Hill, NJ, 07974, USA
Email: {bshep,gtw}@research.bell-labs.com

Abstract

We study networks where end-to-end traffic flow is determined by domains with independent and competing interests. Such *multilateral networks* offer smaller domains the possibility to influence traffic flows if they agree to work together. We explore a cooperative game theoretic model that accounts for such coalitional behaviour. One objective is to model the tradeoffs between transitting third-party traffic versus satisfying one's own traffic demands. In contrast to previous work our focus is on long-term economic interests as opposed to short-term routing decisions, and we do not allow transferrable utility. This leads us to the notion of *harmonious routing*, where each node may only transit as much flow as it offers. This problem has connections to well-known results in optimization including Mader's T-path Theorem, confluent flows, Gomory-Hu trees and a classical result of Scarf in game theory. We end with a summary of many open problems.

Keywords: Interdomain network, cooperative game, combinatorial optimization.

1 Introduction

A *multilateral network* is one that provides end-to-end connectivity not by design of a centralized authority or by mutual agreement on some common metrics, but as the result of various arrangements between the multiple autonomous and self-interested domains that together form the network. The most prominent example has resulted from the opening of the Internet in the 1990's to commercial interests. Operators in such networks need to be able to predict the impact of hardware failures or policy changes within the domain under its control as well as neighbouring domains. Since traditional concepts and methods are not directly applicable, we define a game theoretic model in an attempt to develop a framework to be able to predict such outcomes.

In a multilateral setting there is more opportunity for coalitional behaviour, such as the potential for small operators to affect traffic flow if they agree to work together. In an effort to explain the dynamics of networks carrying each others' traffic, we propose the use of cooperative game theory. We see that the *core* of a certain multilateral transport game is nonempty due to a classical result of Scarf. However, we do not know the complexity of computing such an element. Instead we consider a relaxation called a *harmonious routing*, which is meant to reflect the inherent tradeoffs between satisfying one's own customer demand and transitting third-party traffic. *Transitting* refers to the act of carrying traffic originating at other networks across one's own network, forwarding it towards its ultimate destination. The remainder of the paper explores the complexity of finding such a routing, as well as designing subnetworks that admit one. We close with several open questions.

2 A Multilateral Network Game

A *nontransferrable utility (NTU) coalition game* consists of a set of players V and a mapping Γ from the power set of V, such that for each $S \subseteq V$, $\Gamma(S)$ consists of a subset of $\mathbf{R}^S$ (the space of real vectors whose components are indexed by players in S) of possible payoff vectors to players in S. We also assume that $\{\pi : \pi \in \Gamma(S), \pi_i \geq v_i, \ \forall i \in S\}$ is bounded, where $v_i = \max\{x_i : x \in \Gamma(\{i\})\}$ is the maximum payoff i could receive if she does not join any coalition. Naturally a player will participate in a solution only if his payoff is at least v_i, otherwise they drop (deviate) from the coalition. More generally, the *core* of an NTU coalition game is defined as the set of payoff vectors π in $\Gamma(V)$ with the following property. For each subset $S \subseteq V$, if $y \in \mathbf{R}^S$ and $y_i > \pi_i$ for all $i \in S$, then $y \notin \Gamma(S)$.

For a given coalition game, predicting player behaviours usually involves assuming that the *grand coalition* (i.e., the whole set of players V) is formed and then estimating which payoff vector is adopted. Only payoff vectors in the core make sense if we assume that subsets of players can efficiently negotiate amongst themselves. Many coalition games have an empty core and so proposing a solution to a game sometimes involves examination of approximate core elements.

We now introduce the **multilateral transport game** we study. We are given a (directed or undirected) network $G = (V, E)$ where each node $v \in V$ represents a subnetwork or domain. Each node has some traffic that it wishes to route; throughout we let $D(v)$ denote the total amount of this traffic. In order to balance the cost of transitting others' traffic with the desire to have one's own traffic be routed, we make the following assumptions:

- for each node (subnetwork) v, the utility of carrying v's traffic is λ_v per unit demand
- for each node v, the internal cost of transitting traffic is μ_v per unit demand.

Each node v has a capacity κ_v (and possibly also arcs but we do not consider this case). This is an upper bound on the total traffic handled by that node, i.e., its own traffic plus any transitted traffic. Suppose we are given such a game with the players' demands specified in some form (defined formally later in this section). For any set of nodes S, the set of payoffs $\Gamma(S)$ arises as follows. For any feasible routing of demands between nodes only in S, there is a utility π_v for each node $v \in S$ which is unambiguously defined by the parameters λ, μ applied to the routing.

We now make some remarks on this game. Network operators generally have an estimate on the cost to carry traffic on their network, often on a per-unit basis. They also know how much revenue their own customers generate. For their customer traffic (which terminates at the node) it is reasonable to aggregate the revenue and traffic costs into one parameter λ_v. This is not the case for transitted traffic since it is difficult to estimate the payoff for carrying other networks' traffic. This is in fact the purpose of the model, to measure the tradeoff between the benefits of cooperating and the costs of transitting traffic. This is why we choose to only track costs μ_v of transitting traffic.

Papadimitriou [11] proposed a special case of this game with $\mu_v = 0$ and $\lambda_v = 1$ and each node has a fixed capacity κ_v; in his game, players can also transfer utility between themselves. One notable drawback in this simple case where $\mu_v = 0$ is that κ_v is a constant amount available for any coalition, however small, containing v. It is more realistic that v will dish out its network's capacity to multiple coalitions, and each coalition will be given an amount proportionate to the volume of their traffic handled by the coalition. In particular, since there is no penalty for transitting, it is possible for a node to transit traffic in a core solution even though none of its traffic is being delivered! We offer a different simplification of the multilateral game that avoids these issues.

A *symmetric network game* is an instance where each node's capacity (and hence its leverage) as well as its gain are proportional to the traffic it offers. In other words, we assume that $\lambda_v = 1 = \mu_v$ for all v. While this is a substantial simplification, it is not unreasonable, for instance, that a subnetwork's capacity is dimensioned proportionate to its own customers' traffic.

We now define the routing and flow notation used in the remainder. A *multiflow* in $G = (V, E)$ for a given traffic matrix (D_{ij}), consists of a (fractional) assignment $f : \mathcal{P} \rightarrow \mathbf{R}$ to simple paths in G with the following property. For each $i \neq j$, let $\mathcal{P}_{ij}$ denote the simple paths (directed paths if G is directed) in G with endpoints i and j (source i, destination j if G is directed). We thus have $\mathcal{P} = \cup_{i,j}\mathcal{P}_{ij}$. For a given pair i, j we say that f *fulfills* the pair if $\sum_{P \in \mathcal{P}_{ij}} f(P) = D_{ij}$. For any path P, let $I(P)$ denote the nodes left after deleting the endpoints of P, i.e., P's *internal nodes*. We say that f is *feasible* for capacities κ if for each node v, $\sum_{P:v \in I(P)} f(P) \leq \kappa_v$. Note that the *payoff* to a node v under f is just $\pi_v(f) = \sum_{j \neq v} \sum_{P \in \mathcal{P}_{vj}} \lambda_v f(P) - \sum_{P:v \in I(P)} \mu_v f(P)$. We mention that even though we present our flows in the path formulation (with exponentially many variables) one would obviously compute these flows efficiently using the usual compact formulation and linear programming. If each $f(P)$ is an integer, then we say that f is an *integer multiflow*, or *routing* for the traffic matrix.

For each of these problems we consider two models for demand specification. First, we consider the *traffic matrix model*, where demand is specified by a fixed input traffic matrix (D_{ij}). For such a matrix, we define $D(v) = \sum_j D_{vj}$ to be the marginal demand offered by node v. Second, we consider the *marginal demand model*, where each v specifies only its marginal demand $D(v)$ (or in directed graphs it specifies only its net out-flow). Our objective is to find a routing or multiflow for *some* traffic matrix that meets these marginals.

While the traffic matrix model is more ubiquitous, there are scenarios where marginal demands arise. In some cases, only the marginal demands can be reliably estimated. In emerging telecom markets these are often based on city population growth estimates. A method due to Kruithof estimates future traffic matrices based only on current point-to-point demands and projections of future marginal traffic at each node [7]. Also, if we truly believe that traffic in a network will evolve according to some game, then an alternative approach is to predict future traffic matrices by giving estimates of the marginals and seeing what solutions the game provides.

3 Stable Solutions: The Core

In this section we consider several versions of the multilateral network game just defined. The *multiflow coalition game* is where we are given a network G, demands (D_{ij}) and our parameters λ, μ etc. For each subset S and each multiflow for the demands amongst S, we consider the payoffs for each node in S determined by the parameters λ, μ. Note that we do not require such a multiflow to fulfill all of these demands. Any such flow then yields a payoff vector in $\Gamma(S)$. Similarly, we may define the *routing coalition game*, identical except we only consider integral multiflows.

Figure 1(a) illustrates that a maximum payout-flow is not necessarily in the core. The maximum payout-flow, where all demands are routed, causes the two middle nodes to transit one unit of demand each. By breaking off into a smaller coalition consisting of just themselves, the two middle nodes still get to satisfy all their demand without having to transit any traffic. Even worse perhaps, the core of our game need not even be convex (example omitted), even though each $\Gamma(S)$ is in the case of the multiflow game.

One of the most general results on cores of NTU games is due to Scarf [12]. He defines a collection $\mathcal{C}$ of coalitions as *balanced* if there exists a weight vector $\lambda \geq 0$ such that $\sum_{S \in \mathcal{C}: i \in S} \lambda_S = 1$ for all players i. A vector π can be *achieved* by a set of players S if $\pi^S \in \Gamma(S)$; here, π^S denotes the vector in $\mathbf{R}^S$ obtained by restricting to the components of S. A game is *balanced* if for every balanced collection $\mathcal{C}$ and $\pi \in \mathbf{R}^V$, if π is achieved by each $S \in \mathcal{C}$, then $\pi \in \Gamma(V)$.

Theorem 3.1 (Scarf) *Every balanced NTU coalition game has a non-empty core.*

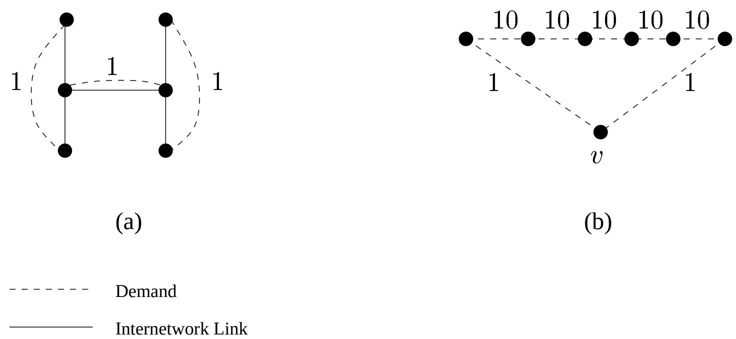

Figure 1: (a) Maximum payout-flow not in the NTU core (b) Tree coalition game with an empty core

One can show that the NTU multiflow coalition game is balanced. The proof of this fact is identical to that for the special case proposed by Papadimitriou [11] where each $\mu_v = 0$ and each node has a fixed capacity κ_v. This was shown by Markakis and Saberi [10]. The following is an immediate consequence.

Corollary 3.2 *The NTU multiflow coalition game has a non-empty core.*

However, we do not know of any polynomial time (polytime) algorithm to compute an element in the core. In fact, this is not resolved even in the case where the network is a tree. If it is a path, however, one may solve for a core element. Namely, it is easy to check if there is a deviation, since there would always be one corresponding to a set of players inducing a connected subpath. Hence given a payout, one may check in polytime whether it is in the core.

For the routing coalition game, the situation is even worse as we do not have any nontrivial conditions that guarantee an element in the core. All of this intractability leads us to propose a relaxation of the core, which is more amenable to computation.

4 Symmetric Games and Harmonious Flows: A Relaxation of the Core

For the remainder, we focus on symmetric network games. It is more tractable, yet still captures many of the impacts of coalitional power structure on the topology and traffic in a network. Based on these assumptions and since we only consider NTU games, each node will transit no more traffic than it routes in any coalition. To this end, we call a routing (or multiflow) *harmonious* if for each node, the amount of flow it transits is bounded by the amount it terminates. If in addition it fulfils each demand, then it is a harmonious routing for the demands. Note that every element in the core of a symmetric multiflow (or routing) coalition game arises from some harmonious multiflow (routing) in G. We also call a graph G *deployable* for the given demands if there exists a harmonious routing (or multiflow in the fractional version) that fulfills each demand. Computing a harmonious multiflow in a graph can be solved by a linear program. The remainder of the paper is focused on deriving efficient techniques for computing harmonious routings. We also consider the problem of finding deployable subgraphs.

4.1 Harmonious Routings

The complexity of finding harmonious routings in the traffic matrix model is always NP-complete. The proofs are omitted due to space.

	Directed harmonious routing	Undirected harmonious routing
traffic matrix	NP-complete (from ARC DISJOINT PATHS)	NP-complete (from SUBGRAPH HOMEOMORPHISM)
marginals	polynomial time	pseudo-polynomial time (polynomial for trees)

Table 1: Harmonious routing complexity results

In the directed setting under the marginal demand model, $D(v)$ is an integer that represents the net flow out of v if it is positive, and otherwise is the net flow into v. Clearly $\sum_v D(v) = 0$ in any feasible instance. Standard network flow algorithms can be used to find a harmonious routing in this case.

For an undirected network under the marginal demand model, one may appeal to a node-capacitated form of Mader's T-path theorem. Given an undirected graph G and stable set (pairwise non-adjacent) of nodes T, a *T-path* is a simple path in G whose endpoints are distinct elements of T. Mader [9] gave a min-max characterization for finding the maximum *internally* disjoint collection of T-paths in a graph. Lovász [8] (cf. [13]) used linear matroid matchings to devise a polynomial time algorithm for finding such T-path packings. In a capacitated form of Mader's theorem, we may ask that each node v outside of T lies in at most some u_v of the T-paths. One checks that a valid reduction is to make u_v copies of v that together form a stable set. Thus there is a pseudo-polynomial time algorithm for the capacitated version. We solve for a harmonious routing in polytime as follows. Let T denote a new stable set of nodes, where for each v, there are $D(v)$ copies, each of which is a leaf hanging from the original $v \in V$. We also give each original node v a capacity of $2D(v)$ of which $D(v)$ is for transitting and $D(v)$ is to receive its flow from the pendant leaves (one small point is that in the capacity reduction we take care to make our leaves only adjacent to the first $D(v)$ copies). We then compute a maximum collection of T-paths. If this collection is of size $\frac{\sum_v D(v)}{2}$ then we are done.

4.2 Harmonious Routing on a Tree

We now consider the special case of finding a harmonious routing on a tree under the marginal demand model if such a flow exists. A pseudo-polynomial time algorithm is obtained by appealing to Mader's Theorem. We now sketch a combinatorial polynomial time algorithm.

We first describe a simple algorithm for (fractional) harmonious multiflows since it forms the basis for the integral case. We first extend our class of problems slightly to the following. We are given a tree $T = (V, E)$ that has a root node r of degree 1 for which $D(r) = 0$. We think of hanging the tree down from r. At the lower levels we label our leaf nodes $v_1, v_2, \ldots, v_l$. Every non-leaf node v has some marginal integer demand value $D(v)$ while each leaf node has integer bounds specified by L_i, U_i. We wish to find a harmonious flow where (1) the total flow from each v_i is an integer in the interval $[L_i, U_i]$ and (2) every other node v admits exactly $D(v)$ amount of traffic. Our initial harmonious flow problem is obtained by setting $L_i = U_i = D(v_i)$ for each leaf node and letting r be a new leaf hanging from the tree.

If the tree is just a star, then the problem amounts to solving a b-matching problem (if there is a solution, we know there is always one where the total flow into each leaf is an integer.) Otherwise, we show how to reduce the problem. Let z be some remote node, i.e., it has exactly one neighbour v^* that is not a leaf; let $v_1, v_2, \ldots, v_s$ be its leaf neighbours. Let $J = \{v^*, z, v_1, v_2, \ldots, v_s\}$ and set $E(J) := \{uw : u, w \in J\}$. Consider an assignment $x : E(J) \to \mathbf{P}_+$ and define the *value* of x as $V(x) := \sum_{u,w} x_{uw}$. In the following we use the standard notation $\delta(v)$ to denote the set of edges incident to a node v. We consider the convex region $P = \{x \in \mathbf{R}_+^{E(J)} : L_i \leq x(\delta(v_i)) \leq U_i$ for each $i, x(\delta(z)) = D(z), V(x) \leq 2D(z)\}$. We let $U^* = \max\{x(\delta(v^*)) : x \in P\}$ and $L^* = \min\{x(\delta(v^*)) : x \in P\}$. These represent the maximum and minimum possible flows into v^* (or along edge zv^* if you prefer) for any harmonious routing on the derived instance on J (so here we think of $L(z) = U(z) = D(z)$ and $L(v^*) = 0, U(v^*) = \infty$). Note that any harmonious flow on the whole tree instance will induce such a vector $x \in P$, where the flow on the edge zv^* is in the range $[L^*, U^*]$.

This observation leads to the following reduction for our problem. Consider shrinking the set $J - v^*$ to a single node z, with the bounds L^*, U^* and suppose this new instance has a solution x'. Then we convert x' into a solution for the original instance as follows. Let $f \in [L^*, U^*]$ (inductively assumed to be integer) be the total flow sent on edges incident to z. By convexity of P, we may also find a vector $x \in P$ with the property that $x(\delta(v^*)) = f$. Then we merge x, x' greedily by matching up the flow $x'(\delta(z))$ with the flow $x(\delta(v^*))$. Clearly all steps may be solved in polynomial time.

This gives the general strategy for finding a harmonious routing. However, the integrality constraint requires some extra technical work. Namely, we consider a more general *parity-constrained* version of the problem where each leaf node v in an instance also comes with a parity $p(v) \in \{0, 1\}$. In a solution, we then require that the total flow terminating at such a leaf v is an integer in the range $[L(v), U(v)]$ which is congruent to $p(v)$ modulo 2. Carrying along this extra information requires us to compute parity constrained b-matchings at each step. The details are omitted due to space.

Finally, we mention that the arguments used above apply equally well to the case where the (transit) capacity on each node v is an arbitrary integer and not just $D(v)$. Hence we can extend the result of [3] for throughput flows on a tree to the more general case of node capacities.

4.3 Deployable Tree Networks

In this section, we ask whether an instance of a symmetric game has a spanning subtree that is deployable. At the end, we also consider a tree coalition game with transferrable utility.

In [5], certain optimal trees are proposed for the creation of virtual private data networks. Constructing virtual multiparty networks may require additional stability conditions such as deployability imposed by the routing game. Interestingly, deployable trees seem to exhibit a center of gravity effect, where certain important domains tend to appear in the middle of the tree. The following table shows that in all but one case, the problem of finding a deployable tree is NP-complete. Proofs are omitted due to space limitations.

	Complete graph	Arbitrary topology
traffic matrix	NP-complete (from PARTITION)	NP-complete (from MINIMUM MULTIWAY CUT)
marginals	polynomial time	NP-complete (from PARTITION)

Table 2: Deployable subtree results

We now consider the *greenfield* design problem, that is, where we are designing a network from scratch. Equivalently, we are given a complete graph and marginal demands and we must determine if there is a deployable tree and if so, to produce such a tree and a harmonious routing.

Theorem 4.1 *A set of marginal demands $\{D(v) : v \in V\}$ admits a deployable tree if and only if $\sum_v D(v)$ is even and $D(v) \leq \sum_{u \neq v} D(u)$ for each node v. Moreover, if such a tree exists, then it can be computed in $O(n \log n)$ time and has the property that at most one node transits traffic ($n = |V|$).*

Proof: Order the nodes $v_1, v_2, \ldots, v_n$ so that $D(v_i) \leq D(v_{i+1})$ for each $i < n$. We proceed iteratively and at each step we have a modified demand set $D'(v_i)$ that satisfies our original hypotheses and also that $D'(v_i) \leq D'(v_{i+1})$. Notice that in order to show that $D'(v) \leq \sum_{u \neq v} D'(u)$ it suffices to show that $D'(v_n) \leq \sum_{j<n} D'(v_j)$. To start $D'(v_i) = D(v_i)$ of course. Note that if $D(v_n) = \sum_{i<n} D(v_i)$ then there is a deployable tree obtained by making a star centered at v_n. So we assume this is not the case, and we repeat the following process for $i = 1, 2, \ldots$. Add the edge $v_i v_{i+1}$ to the tree. Set $\Delta = \frac{(\sum_{j<n} D'(v_j) - D'(v_n))}{2} > 0$. Now route $\delta = \min\{D'(v_i), \Delta\}$ on the new edge and reduce each of $D'(v_i), D'(v_{i+1})$ by this amount. Note that all three of our required conditions remain valid for the resulting D'. If $D'(v_i) = 0$, then we continue. Otherwise, we have that $D'(v_n) = \sum_{j<n} D'(v_j)$ and so we add the edges $v_j v_n$ for each $j > i$, and route $D'(v_j)$ on edge $v_j v_n$ for each $j > i$, and in addition route $D'(v_i)$ on the path $v_i v_{i+1} v_n$. This is a harmonious routing since $D'(v_{i+1}) \geq D'(v_i)$. □

We close by considering the deployable tree problem as a *tree coalition game*. In this example, however, we allow transferrable utility (TU). That is, players may share some of their payoff with others in order to keep them in the coalition. Thus a coalition only needs to maximize its total utility, called its *value* $\eta(V)$. We now see that this may be done by means of the Gomory-Hu tree. A sub-coalition deviates in a TU game if it can generate a greater total utility for itself.

Given an undirected complete graph G, for each subset S we let $D(S) = \sum_{ij} D_{ij}$ denote the total utility for carrying traffic within S. For S to maximize its total utility, we need a spanning tree in $G[S]$ that maximizes $D(S)$ minus the total cost to transit traffic. This determines the value of the coalition S. Since $D(S)$ is fixed, this is equivalent to minimizing $\sum_{i,j \in S} D_{ij}(dist_T(i,j) - 1) = \sum_{i,j \in S} D_{ij} dist_T(i,j) - D(S)$. As shown by [6], this is minimized by the Gomory-Hu tree on the complete graph on S with edge weights (D_{ij}). Efficient combinatorial methods are known for computing such a tree [2, 4].

In the tree coalition game, one has that $\eta(S) \geq 0$ for each S. Indeed for any such set, pick any node $v \in S$ and create a star-tree centered at v. One easily checks that the transit cost of this tree is $\sum_{i,j \in S-v} D_{ij} + D_{ji}$ which is at most $D(S)$. Since the total gain is $D(S)$, utility can be transferred so that each node stands to gain. This game may have an empty core however, as Figure 1(b) shows. The figure indicates the demands between nodes. Note that $\eta(V - v)$ is 100 since we can route all demands in $V - v$ on a tree without transitting. However, one checks that any spanning tree for the grand coalition has a total utility of less than 100.

5 Open Problems

We summarize some of the open problems we encountered. (1) What is the complexity of computing an element of the core of a NTU multiflow coalition game? (2) Determine the bounds on the degree of the inapproximability of the stated NP-complete problems. (3) Is the capacitated T-Path problem (or the undirected harmonious routing problem) polynomially solvable? and (4) When does the TU tree coalition game have a nonempty core?

Consider a *single source multilateral transport game* where for each node v, other than a root r, and for each path P from v to r, there is a number $u(P)$ defined as v's utility for using P. In addition, each node v has a transit cost μ_v of forwarding a packet to r, and a total demand $D(v)$ of packets it needs to send to r. Consider a given tree T rooted at r, and a node v whose path to r in T is P. Let $T(v)$ be those nodes, other than v, whose path to r goes through v. Then v's utility $U(T, v)$ is $u(P) - \sum_{x \in T(v)} D(x)\mu_v$. This problem includes as a special case the *confluent flow problem.* Here we are given a directed graph D with a root node r. Let $v_1, v_2, \ldots, v_d$ be the in-neighbours of r. Let T be an arborescence rooted at r. Consider routing $D(v)$ units of flow from each node v to r on the unique path in T. For each i, let l_i, denote the total flow on the arc (v_i, r). The *load* of T is then the maximum of these l_i's. The objective is to find T with a minimum load. Recently, [1] showed that it is NP-hard to approximate the optimal load to within an $O(\log(n))$ factor and that this factor is achievable in polynomial time. We ask whether similar approximations can be obtained for our transport game; specifically for the case where each node has $\mu_v = 0, \lambda_v = 1$ and is endowed with some capacity κ_v.

Acknowledgements: We are grateful to an anonymous referee for a number of excellent suggestions.

References

[1] J. Chen, R. D. Kleinberg, L. Lovasz, R. Rajaraman, R. Sundaram, and A. Vetta. (Almost) tight bounds and existence theorems for confluent flows. In *Proc. of ACM STOC*, pages 529–538, 2004.

[2] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization, New York, 1998.

[3] M.C. Costa, L. Létocart, and F. Roupin. Polynomial algorithms to solve the multiway cut and integer flow problems in trees. *OR Letters*, 31(1):21–27, 2003.

[4] R.E. Gomory and T.C. Hu. Multi-terminal network flows. *J. SIAM*, 9:551–570, 1961.

[5] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: A network design problem for multicommodity flows. In *Proc. of ACM STOC*, pages 389–398, 2001.

[6] T.C. Hu. Optimum communication spanning trees. *SIAM Journal on Computing*, 3:188–195, 1974.

[7] J. Kruithof. Telefoonverkeersrekening. *De Ingenieur*, 52(8):E15–E25, 1937.

[8] L. Lovász. Matroid matching and some applications. *J. of Comb. Theory, Series B*, 28:208–236, 1980.

[9] W. Mader. Über die maximalzahl kreuzungsfreier H-wege. *Archiv der Mathematik*, 31:387–402, 1978.

[10] E. Markakis and A. Saberi. On the core of the multicommodity flow game. In *Proc. of EC*, pages 93–97, 2003.

[11] C. Papadimitriou. Algorithms, games, and the internet. In *Proc. of ACM STOC*, pages 749–753, 2001.

[12] H. Scarf. The core of an n person game. *Econometrica*, 35:50–69, 1967.

[13] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*. Springer, Heidelberg, Germany, 2003.