Bicriteria Network Design via Iterative Rounding

Piotr Krysta⋆

Department of Computer Science, Dortmund University
Baroper Str. 301, 44221 Dortmund, Germany
piotr.krysta@cs.uni-dortmund.de

Abstract. We study the edge-connectivity survivable network design problem with an additional linear budget constraint. We give a strongly polynomial time $(3,3)$-approximation algorithm for this problem, by extending a linear programming based technique of iterative rounding. Previously, a $(4,4)$-approximation algorithm for this problem was known. The running time of this previous algorithm is not strongly polynomial.

1 Introduction

Let $G = (V, E)$ be an undirected multi-graph with cost $c_e \geq 0$ on each edge $e \in E$. For $S \subset V$ and $E' \subseteq E$, we denote by $\delta_{G'}(S)$ the set of edges with one end in S and the other end in $V \setminus S$ in the graph $G' = (V, E')$. We omit the subscript G' in $\delta_{G'}(S)$, if $E' = E$ or if E' is clear from the context.

Edge-Connectivity Survivable Network Design Problem (EC-SNDP) is, given integer requirements r_{uv} for $u, v \in V$, $u \neq v$, to find a minimum-cost subgraph, with respect to costs c_e, containing for any u, v at least r_{uv} edge-disjoint u-v paths. EC-SNDP can be formulated as the following integer linear program (ILP), which we will call (IP1):

$$\begin{aligned} \min \quad & \textstyle\sum_{e \in E} c_e x_e & \\ s.t. \; & \textstyle\sum_{e \in \delta_G(S)} x_e \geq f(S) \; \forall S \subseteq V & (1)\\ & x_e \in \{0,1\} \qquad \forall e \in E, & (2) \end{aligned}$$

where $f(S) = \max\{r_{uv} : u \in S, v \notin S\}$.

The defined problem generalizes many known graph optimization problems, e.g., the Steiner tree problem [10], Steiner forest problem [1, 3], minimum-cost k-edge-connected spanning subgraph problem [5], and others. In a breakthrough paper, Jain [4] has designed a 2-approximation algorithm for EC-SNDP by employing a technique of iterative rounding.

⋆ The author is supported by DFG grant Kr 2332/1-1 within Emmy Noether program. This work has been partially done while visiting the Department of Computer Science of RUTGERS–The State University of New Jersey, Camden, NJ, U.S.A., May 2004.

When designing networks it is often natural to seek for a cheap network as in EC-SNDP that simultaneously optimizes some additional linear cost objective– a so-called budget constrained problem. Such bicriteria optimization problems have been studied by many researchers, see, e.g., [2, 6, 8, 9].

We now define a bicriteria version of EC-SNDP. Given an additional cost (length) $l_e \geq 0$ on each $e \in E$, and a number $L > 0$, we consider EC-SNDP with an additional budget constraint, $\sum_{e \in E} l_e x_e \leq L$, called Constr EC-SNDP.

Let now $opt(G)$ denote the cost of an optimum solution to a given instance G of Constr EC-SNDP, e.g., the solution with minimum cost with respect to c among all solutions with length at most L. An (α, β)-approximation algorithm for Constr EC-SNDP is a polynomial time algorithm that always finds a solution G' to the instance G of this problem, such that G' has cost at most $\alpha \cdot opt(G)$ with respect to c, and such that G' has length at most $\beta \cdot L$.

Results of Marathe et al. [6] and the 2-approximation algorithm for EC-SNDP by Jain [4] imply a $(4, 4)$-approximation algorithm for Constr EC-SNDP. This approach uses a parametric binary search and the 2-approximation algorithm as a subroutine. Thus, the running time of the resulting $(4, 4)$-approximation algorithm is not strongly polynomial. We extend the technique of iterative rounding and obtain an improved $(3, 3)$-approximation algorithm for Constr EC-SNDP. Our algorithm additionally has strongly polynomial running time.

2 A polyhedral theorem

This section contains the main technical ingredient of our method. To prove it, i.e., to prove Theorem 2, we need some new notions and notations – we will mostly rely on Jain's paper [4] in this respect. Nevertheless, we will define below the most important notions from [4]. The proof below also requires some basic knowledge in linear programming and polyhedral combinatorics, see, e.g., [11]. A set function $f : 2^V \to \mathcal{Z}$ is *weakly supermodular* if:

1. $f(V) = 0$, and
2. For every $A, B \subseteq V$ at least one of the following holds:
   (a) $f(A) + f(B) \leq f(A \setminus B) + f(B \setminus A)$,
   (b) $f(A) + f(B) \leq f(A \cap B) + f(A \cup B)$.

We note, that the specific set function $f(S) = \max\{r_{uv} : u \in S, v \notin S\}$ defined before for EC-SNDP is weakly supermodular, see [4]. Let $\Pi = \Pi(E)$ correspond to constraints (1)–(2), and Π' be the linear programming relaxation of Π, that is, Π' is Π with constraints (2) replaced by $x_e \in [0, 1]$ $\forall e \in E$. Thus, Π' is the EC-SNDP polytope. Given an $x \in \Pi'$, i.e., a feasible fractional solution to the EC-SNDP relaxation Π' and a set $S \subseteq V$, we define $x(\delta(S)) = \sum_{e \in \delta(S)} x_e$. For a subset $S \subseteq V$, let $\boldsymbol{x}(S) \in \{0, 1\}^{|E|}$ be the 0-1 vector of the coefficients in the left-hand-side in constraint (1). Having fixed an $x \in \Pi'$ and $S \subseteq V$, we say that set S (or the corresponding constraint (1)) is *tight* if $x(\delta(S)) = f(S)$. Given a family of sets $\mathcal{T} \subseteq 2^V$, let $sp(\mathcal{T})$ denote the vector space spanned by the vectors $\{\boldsymbol{x}(S) : S \in \mathcal{T}\}$. Finally, we say that two sets $A, B \subseteq V$ *cross* if none of the sets

$A \setminus B$, $B \setminus A$ and $A \cap B$ is empty. A family of sets is *laminar* if no two sets in it cross. Each given edge $e = (u, v) \in E$ has two *endpoints*, denoted by e_u and e_v. Since we have $|E|$ edges, we have exactly $2 \cdot |E|$ endpoints.

2.1 A first polyhedral theorem

We first show that even without going into the combinatorial structure of the Constr EC-SNDP polytope we are able to easily obtain a $(4, 4)$-approximation algorithm with strongly polynomial running time. Namely, we shall prove the following fact.

Theorem 1. *Let x^* be any extreme point of the EC-SNDP polytope defined by a weakly supermodular function f with an additional linear constraint lin. Then there is an index $e \in E$ such that $x^*_e \geq 1/4$.*

Proof. Let x^* be a given extreme point of the polytope as in the theorem. We will use the following well known characterization of extreme points (see, e.g., page 104 in the book by Schrijver [11]).

Lemma 1. *Let $P = \{x \in \Re^m : Ax = a, Dx \leq d\}$ be a polytope defined by a system $Ax = a, Dx \leq d$ of linear equations and inequalities. Then $\bar{x} \in P$ is an extreme point of P if and only if there exists a set of m equations and tight inequalities from the system defining P, such that $\bar{x}$ is the unique solution to the corresponding equation system.*

Thus, x^* is determined by $|E| = m$ linearly independent equations, say $\mathcal{S}^*$, from the system $\Pi' \cup \{lin\}$. (Let us recall, that Π' corresponds to constraints (1)–(2), with constraints (2) replaced by $x_e \in [0, 1]\ \forall e \in E$.)

Assume first that constraint lin is present as a tight equation in system $\mathcal{S}^*$. If we skip the equation corresponding to constraint lin from $\mathcal{S}^*$, we are remaining with a new system, say $\mathcal{S}^*_1$. Since $\mathcal{S}^*_1$ has $m - 1$ linearly independent equations, it defines a 1-dimensional affine subspace, say F, of the polytope Π'. We know that $x^* \in F$. Therefore, x^* belongs to the intersection of F and the hyperplane defined by constraint lin. Since F is a 1-dimensional face of polytope Π', we can decompose x^* into a convex combination of at most 2 extreme points of polytope Π' (this fact basically follows from the theorem of Carathéodory, and the fact that any point in a polytope is a convex combination of its vertices, see [11]). If constraint lin is not present as a tight equation in system $\mathcal{S}^*$, then x^* is an extreme point of the polytope Π' itself. Thus, we have shown the following lemma.

Lemma 2. *Any extreme point x^* of the EC-SNDP polytope defined by a weakly supermodular function f with an additional linear constraint lin can be expressed as a convex combination of at most 2 extreme points of the EC-SNDP polytope without constraint lin.*

But we know by the result of Jain [4], that any extreme point of polytope Π' has at least one entry of at least 1/2. We will use this fact to proceed. By Lemma 2, $x^* = \lambda x^1 + (1-\lambda)x^2$, where $\lambda \in [0,1]$, and x^1, x^2 are extreme points of the polytope Π'. Then, either $\lambda \geq 1/2$ or $1-\lambda \geq 1/2$. Suppose without loss of generality that $\lambda \geq 1/2$. From the result of Jain mentioned above, we know that there exists an $e \in E$, with $x^1_e \geq 1/2$. Therefore, we obtain that

$$x^*_e = \lambda x^1_e + (1-\lambda)x^2_e \geq \lambda x^1_e \geq 1/4.$$

This concludes the proof of Theorem 1. □

2.2 An improved polyhedral theorem

We will now show how to improve the result of Theorem 1 by exploring the combinatorial structure of the Constr EC-SNDP polytope.

Theorem 2. *Let x^* be any extreme point of the EC-SNDP polytope defined by a weakly supermodular function f with an additional linear constraint lin. Then there is an index $e \in E$ such that $x^*_e \geq 1/3$.*

Proof. This proof builds on some arguments that have been already used by Jain [4]. We assume w.l.o.g. that for any $e \in E$, $0 < x^*_e < 1$. Otherwise, we either can project the polytope onto an $x^*_e = 0$ or we are done if $x^*_e = 1$. Let $\mathcal{S}^*$ be a system defining the extreme point x^*, see Lemma 1. If constraint lin is not tight, then x^* is an extreme point of the EC-SNDP polytope, so by result of Jain [4], there exists an $e \in E$ such that $x^*_e \geq 1/2$, and we are done.

Assume constraint lin is tight for x^* and it is present as a tight equation in the system $\mathcal{S}^*$. Let $\mathcal{T}$ be a family of all tight sets for x^*, i.e.,

$$\mathcal{T} = \{S \subset V : x^*(\delta(S)) = f(S)\}.$$

Let $\mathcal{L} \subseteq \mathcal{T}$ be a maximal laminar subfamily of $\mathcal{T}$. By Lemma 4.2 in [4], $sp(\mathcal{L}) = sp(\mathcal{T})$. We claim that the vector space $sp(\mathcal{T})$ has dimension $\geq |E| - 1 = m - 1$. This is clear, since

$$dim(sp(\mathcal{T}) \cup \boldsymbol{x}_{lin}) = m,$$

where $\boldsymbol{x}_{lin}$ is the vector of the coefficients of the left-hand-side of lin. If now $dim(sp(\mathcal{T})) = m$ then by Jain's [4] theorem there is an e with $x^*_e \geq 1/2$, so the theorem holds. So assume $dim(sp(\mathcal{T})) = m - 1$. Since $sp(\mathcal{L}) = sp(\mathcal{T})$, there is a basis $\{\boldsymbol{x}(S) : S \in \mathcal{B}\}$ of the vector space $sp(\mathcal{T})$, with $\mathcal{B} \subseteq \mathcal{L}$ and $|\mathcal{B}| = m - 1$. We also notice that for each $S \in \mathcal{B}$ we have $f(S) \geq 1$. This fact follows by a contradiction. Namely, suppose that $f(S) < 0$, then S cannot be tight; also if $f(S) = 0$, then $\boldsymbol{x}(S) = \mathbf{0}$, which gives a contradiction to linear independence. The reasoning above leads therefore to the following lemma.

Lemma 3. *There is a laminar family $\mathcal{B}$ of tight sets, such that $|\mathcal{B}| = m - 1$, vectors $\boldsymbol{x}(S)$, $S \in \mathcal{B}$ are linearly independent, and $\forall S \in \mathcal{B}$, $f(S) \geq 1$.*

We represent the laminar family $\mathcal{B}$ as a directed forest as follows. The node set of the forest is $\mathcal{B}$, and there is an edge from $C \in \mathcal{B}$ to $R \in \mathcal{B}$ if R is the smallest set containing C. We say that R is a parent of C and C is a child of R. A parent-less node is called a root, and a child-less node is called a leaf. We prove now the following lemma that will imply the theorem.

Lemma 4. *Laminar family $\mathcal{B}$ contains at least one set S with $|\delta(S)| \leq 3$.*

Proof. Assume towards a contradiction that for any set $S \in \mathcal{B}$ we have $|\delta(S)| \geq 4$. We show a contradiction by distributing the $2|E| = 2m$ endpoints to the $m-1$ nodes of the forest representing the laminar family $\mathcal{B}$. In this distribution, each internal node of a subtree gets at least 2 endpoints, and the root gets at least 4 endpoints. This is true for all subtrees of the forest that are leaves: each such leaf S is crossed by at least 4 edges. Induction step is the same as in the proof of Lemma 4.5 in [4], and we present it here for completeness.

Consider a subtree rooted at R. Suppose first R has two or more children. By the induction hypothesis, each of those children gets at least 4 endpoints, and each of their descendants gets at least 2 endpoints. We will now redistribute these endpoints. Root R borrows two endpoints from each of its children – thus getting at least 4 endpoints. Hence, each descendant of R will still have at least 2 endpoints, and we are done in this case. Assume now, that R has exactly one child, say C. By the induction hypothesis, C gets at least 4 endpoints, and each of its descendants gets at least 2 endpoints. We now borrow two endpoints from C and assign them to R. If R had two more endpoints on its own, i.e., endpoints which were incident to R, the induction step follows. Thus, the only case that is left is to assume that R has at most one endpoint incident to it.

Since $\boldsymbol{x}(R)$ and $\boldsymbol{x}(C)$ are two distinct vectors, there is at least one edge which crosses C but not R, or else crosses R but not C. In both cases there is an endpoint incident to R. By our assumption R has at most one endpoint incident to it, and so R has exactly one endpoint incident to it.

The value x_e of the edge giving one endpoint incident to R is the difference between the requirements of R and C. This is an integer, but by our assumption x_e is strictly fractional – a contradiction. This proves the induction step.

If the forest has at least two roots, we get a contradiction, since the number of endpoints is $2m$, but by the distribution it is at least $2(m-1)+4 = 2m+2$.

Thus, we assume that the forest has exactly one root. In this case the forest is just a tree. We know that after the distribution the (unique) root of the tree has got 4 endpoints. We observe that the root corresponds to a set R of vertices of the original graph, such that R contains every other set in the laminar family $\mathcal{B}$ (by the uniqueness of the root). But by our assumption R must be crossed by at least 4 edges. This means that there are at least 4 endpoints (corresponding to the ends of these 4 edges that are outside R), and we know that these four endpoints were not considered so far in the distribution (since they are not contained in the tree). Therefore we can assign these 4 additional endpoints to the root R. Thus the root gets at least 8 endpoints. This means that the number of endpoints is at least $2(m-1)+6 = 2m+4$. Contradiction. This finishes the proof of Lemma 4, and the proof of Theorem 2, as well. $\square$

3 The algorithm

Let $G = (V, E)$ be a given undirected multigraph, with a cost function $c : E \to \Re_+$ and a length function $l : E \to \Re_+$. Let L be a given positive number. Assume that problem $IP(\Pi)$ is to find a minimum-cost (with respect to c) subgraph G' of G such that (1) and (2) hold, and cost of G' with respect to l is at most L. Let $\Pi = \Pi(E)$ correspond to constraints (1)–(2). We can formulate $IP(\Pi)$ as:

$$\min \quad \textstyle\sum_{e \in E} c_e x_e \tag{3}$$
$$s.t. \quad x \in \Pi(E) \tag{4}$$
$$\textstyle\sum_{e \in E} l_e x_e \leq L \tag{5}$$
$$x_e \in \{0, 1\} \quad \forall e \in E. \tag{6}$$

Let (LP) denote the LP relaxation of $IP(\Pi)$, that is, (LP) corresponds to (3)–(6) with (6) replaced by $x_e \in [0, 1]$ $\forall e \in E$.

The algorithm is recursive. In the first step it solves the (LP) optimally, producing a basic solution $\bar{x}$ to (LP). By Theorem 2 there exists $e \in E$ such that $\bar{x}_e \geq \frac{1}{3}$. Let $E(\frac{1}{3}) = \{e \in E : \bar{x}_e \geq \frac{1}{3}\}$, and opt_{LP} denote the value of an optimal fractional solution to (LP).

The algorithm rounds each $\bar{x}_e$, for $e \in E(\frac{1}{3})$, to one: $E(\frac{1}{3})$ is a part of the solution. Then the algorithm recursively solves the following integer program, called (IP$'$).

$$\begin{array}{cc} \min & \sum_{e \in E \setminus E(\frac{1}{3})} c_e x_e \\ s.t. & x \in \Pi(E \setminus E(\frac{1}{3})) \\ & \sum_{e \in E \setminus E(\frac{1}{3})} l_e x_e \leq L - \sum_{e \in E(\frac{1}{3})} l_e \bar{x}_e \\ & x_e \in \{0, 1\} \quad \forall e \in E \setminus E(\frac{1}{3}), \end{array}$$

where $\Pi(E \setminus E(\frac{1}{3}))$ corresponds to constraints (1)–(2) with E replaced by $E \setminus E(\frac{1}{3})$, and constraints (1) replaced by

$$\sum_{e \in \delta_{G'}(S)} x_e \geq f(S) - |E(1/3) \cap \delta_G(S)| \quad \forall S \subseteq V, \quad \text{where} \quad G' = (V, E \setminus E(1/3)).$$

We summarize our algorithm for Constr EC-SNDP in Figure 1 below.

We would like to note here that if we use in the algorithm above Theorem 1 instead of Theorem 2 this would obviously lead to a weaker approximation algorithm for Constr EC-SNDP. However, the arguments in the proof of Theorem 1 may prove useful – see Section 5.

4 The analysis

We will first prove the following fact about all linear programs that may appear during the execution of the algorithm.

Algorithm **IterativeRound**(E, L)**:**

if E has constant size **then** solve E optimally and output its solution;

solve (LP) optimally; let $\bar{x} \in [0,1]^{|E|}$ be the resulting basic solution;

$E(\frac{1}{3}) := \{e \in E : \bar{x}_e \geq \frac{1}{3}\}$;

output $E(\frac{1}{3}) \cup$ **IterativeRound**$(E \setminus E(\frac{1}{3}),\ L - \sum_{e \in E(\frac{1}{3})} l_e \bar{x}_e)$.

Fig. 1. The recursive iterative rounding algorithm for Constr EC-SNDP.

Lemma 5. *All the linear programs of form (LP) as defined in Section 3 that the algorithm IterativeRound will meet during its execution are fractionally feasible.*

Proof. The proof is by induction on the iterations of the algorithm. Let us consider the first iteration, in which the algorithm solves exactly (LP) as above. Then, since our original integral problem Constr EC-SNDP has an integral feasible solution, this solution is also feasible for the (LP).

We will now show the induction step. Let (LP′) be the LP relaxation of (IP′), obtained in the same way as (LP). Suppose now that (LP) was a linear program met by our algorithm in some iteration, and let (LP′) be the linear program met in the very next iteration. By induction assumption (LP) is fractionally feasible, and let $\bar{x}$ be an optimal basic fractional solution to (LP). The restriction of $\bar{x}$ to $E \setminus E(\frac{1}{3})$ is a feasible solution to (LP′), and so the claim holds. □

We are now ready to prove the main theorem in this paper.

Theorem 3. *The algorithm IterativeRound defined in Section 3 is a strongly polynomial time* $(3, 3)$*-approximation algorithm for the* Constr EC-SNDP *problem.*

Proof. We will prove the following statement: if a linear program (LP) as defined in Section 3 is met by the algorithm at some iteration and this linear program is fractionally feasible, then our algorithm outputs a corresponding integral solution to this (LP) which is within a factor of 3 of the optimal fractional solution to this (LP), and the length of this integral solution is within a factor of 3 of the budget in this (LP). We will prove this statement by induction on the number of iterations of our algorithm.

Suppose first that an instance of Constr EC-SNDP requires only one iteration of the algorithm. By Lemma 5, the linear program involved, say (LP), is fractionally feasible. Thus, the algorithm finds an optimum basic solution to (LP), say $\bar{x}$, and rounds up all edges e with $\bar{x}_e \geq 1/3$. Thus, $E(\frac{1}{3})$ is an integral feasible solution to (LP), and we will argue that it gives a $(3, 3)$-approximation. Since for each $e \in E(\frac{1}{3})$, $3\bar{x}_e \geq 1$, we have that

$$\sum_{e \in E(\frac{1}{3})} c_e \leq 3 \cdot opt_{LP}.$$

Similarly, we can argue that

$$\sum_{e \in E(\frac{1}{3})} l_e \leq \sum_{e \in E(\frac{1}{3})} 3\bar{x}_e l_e \leq 3 \sum_{e \in E} \bar{x}_e l_e \leq 3L.$$

We will now show the induction step. Observe, that the LP which is the relaxation of (IP′) still fulfills Theorem 2 (this follows from Theorem 2.5 in [4]). Thus, (IP′) is an instance of the same form as (IP). Let (LP′) be the LP relaxation to (IP′), obtained in the same way as (LP). We now argue about the approximation ratio. By Lemma 5 the linear program (LP′) is fractionally feasible, and so by induction assumption there is an integral solution, say E', to (LP′) with cost at most $3 \cdot opt_{LP'}$ and length at most

$$3 \cdot \left(L - \sum_{e \in E(1/3)} l_e \bar{x}_e \right).$$

We show that $E(\frac{1}{3}) \cup E'$ is an integral solution to (LP) with cost at most $3 \cdot opt_{LP}$ and length at most $3L$.

The restriction of $\bar{x}$ to $E \setminus E(\frac{1}{3})$ is a feasible solution to (LP′), so

$$opt_{LP'} \leq opt_{LP} - \sum_{e \in E(\frac{1}{3})} c_e \bar{x}_e.$$

But for each $e \in E(\frac{1}{3})$, $3\bar{x}_e \geq 1$, hence

$$3 \cdot opt_{LP'} + \sum_{e \in E(\frac{1}{3})} c_e \leq 3 \cdot opt_{LP}.$$

From the induction assumption, $\sum_{e \in E'} c_e \leq 3 \cdot opt_{LP'}$, thus

$$\sum_{e \in E'} c_e + \sum_{e \in E(\frac{1}{3})} c_e \leq 3 \cdot opt_{LP}.$$

Also by the induction assumption, $\sum_{e \in E'} l_e \leq 3 \cdot (L - \sum_{e \in E(\frac{1}{3})} l_e \bar{x}_e)$. Therefore, the length of the output solution is

$$\sum_{e \in E'} l_e + \sum_{e \in E(\frac{1}{3})} l_e \leq \sum_{e \in E'} l_e + 3 \cdot \sum_{e \in E(\frac{1}{3})} l_e \bar{x}_e \leq 3 \cdot L.$$

Jain [4] has shown that LPs (LP) corresponding to our original problem without the budget constraint can be solved in strongly polynomial time. Norton et al. [7] prove that if we have a strongly polynomial time algorithm for a linear program, then this linear program can still be solved in strongly polynomial time when a constant number of linear constraints are added to the linear program. This gives a strongly polynomial algorithm for our problem. (The approach in [7] requires in fact a specific strongly polynomial time algorithm, called "linear", and [4] indeed provides such an algorithm.)

We will finally note that we could also use Theorem 1 instead of Theorem 2 to obtain a strongly polynomial time $(4, 4)$-approximation algorithm for the Constr EC-SNDP problem. □

5 Conclusions and further questions

We have obtained in this paper a strongly polynomial time $(3, 3)$-approximation algorithm for the Constr EC-SNDP problem, by extending the iterative rounding technique. For this result we needed explore the combinatorial structure of the Constr EC-SNDP polytope in Section 2.2.

We have also shown in Section 2.1 that a straightforward polyhedral argument implies a weaker strongly polynomial time $(4, 4)$-approximation algorithm for this problem. This second polyhedral argument might be interesting for the following reason. Suppose, that we are given a problem which can be formulated as an integer linear program. Let us now take an LP relaxation to this integer linear program, and suppose that we are able to prove that any two adjacent extreme points of this relaxation have two large entries (maybe even at the same vector position). Then, we will be able to show that an extreme point of this polytope with an additional linear constraint also has a large entry – see the argument at the end of the proof of Theorem 1. Similar argument may be applied in case where we add more than just one additional linear constraint. We leave finding such applications as an open problem.

References

1. A. Agrawal, P. Klein and R. Ravi. When trees collide : An approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, **24**(3), pp. 445–456, 1995.
2. J.L. Ganley, M.J. Golin and J.S. Salowe. The Mutli-Weighted Spanning Tree Problem. In the *Proc. 1st Conference on Combinatorics and Computing (COCOON)*, Springer Verlag, LNCS, pp. 141–150, 1995.
3. M.X. Goemans and D.P. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM Journal on Computing*, **24**, pp. 296–317, 1995.
4. K. Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica*, **21**(1), pp. 39–60, 2001. Available at http://www.cc.gatech.edu/people/home/kjain/
5. S. Khuller and U. Vishkin. Biconnectivity approximations and graph carvings. *Journal of the ACM*, **41**, pp. 214–235, 1994.
6. M.V. Marathe, R. Ravi, R. Sundaram, S.S. Ravi, D.J. Rosenkrantz and H.B. Hunt III. Bicriteria Network Design. *Journal of Algorithms*, **28**, pp. 142–171, 1998.
7. C.H. Norton, S.A. Plotkin and É. Tardos. Using Separation Algorithms in Fixed Dimension. *Journal of Algorithms*, **13**, pp. 79–98, 1992.
8. R. Ravi and M.X. Goemans. The Constrained Minimum Spanning Tree Problem. In the *Proc. of the Scandinavian Workshop of Algorithmic Theory (SWAT)*, Springer Verlag, LNCS **1097**, pp. 66–75, 1996.
9. R. Ravi, M.V. Marathe, S.S. Ravi, D.J. Rosenkrantz and H.B. Hunt III. Many birds with one stone: multi-objective approximation algorithms. In the *Proc. 25th ACM Symposium on Theory of Computing (STOC)*, 1993.
10. G. Robins and A. Zelikovsky. Improved Steiner Tree Approximation in Graphs. In the *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 770–779, 2000.
11. A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, Chichester, 1986.