REVUE D'ANALYSE NUMÉRIQUE ET DE THÉORIE DE L'APPROXIMATION
Rev. Anal. Numér. Théor. Approx., vol. 34 (2005) no. 1, pp. 93–102
ictp.acad.ro/jnaat

APPROXIMATION THEORY IN COMBINATORIAL OPTIMIZATION. APPLICATION TO THE GENERALIZED MINIMUM SPANNING TREE PROBLEM

PETRICA C. POP*, G. STILL† and W. KERN†

Abstract. We present an overview of the approximation theory in combinatorial optimization. As an application we consider the Generalized Minimum Spanning Tree (GMST) problem which is defined on an undirected complete graph with the nodes partitioned into clusters and non-negative costs are associated to the edges. This problem is NP-hard and it is known that there cannot exists a polynomial approximation algorithm. We present an in-approximability result for the GMST problem and under special assumptions: cost function satisfying the triangle inequality and with cluster sizes bounded by ρ, we give an approximation algorithm with ratio 2ρ.

MSC 2000. 90C11, 90C27, 05C05.

Keywords. Generalized minimum spanning tree problem.

1. APPROXIMATION THEORY IN COMBINATORIAL OPTIMIZATION

Combinatorial Optimization is the process of finding one or more best (optimal) solutions in a well defined discrete problem space, i.e. a space containing a finite set of possible solutions, that optimizes a certain function, the so-called *objective function*. The finite set of possible solutions can be described by inequality and equality constraints, and by integrality constraints. The integrality constraints force the variables to be integers. The set of points that satisfy all these constraints is called the *(feasible) solution set*.

Such problems occur in almost all fields of management (e.g. finance, marketing, production, scheduling, inventory control, facility location, etc.), as well as in many engineering disciplines (e.g. optimal design of waterways or bridges, design and analysis of data networks, energy resource-planning models, logistic of electrical power generation and transport, etc.). A survey of applications of combinatorial optimization is given by Grötschel in [7].

*Department of Mathematics and Computer Science, Faculty of Sciences, North University of Baia Mare, Romania, e-mail: pop_petrica@yahoo.com.

†Faculty of Mathematical Sciences, University of Twente, P.O. Box 217, 7500 Enschede, the Netherlands e-mail: {w.kern,g.still}@math.utwente.nl.

Combinatorial Optimization models are often referred to as integer programming models where some or all of the variables can take on only a finite number of alternative possibilities.

Many of the optimization problems we would like to solve are $\mathcal{NP}$-*hard*. Therefore it is very unlikely that these problems could be solved by a polynomial-time algorithm. However, these problems still have to be solved in practice. In order to do that, we have to relax some of the requirements. There are in general, three different possibilities.

- **Superpolynomial-time algorithms:** Even though an optimization problem is $\mathcal{NP}$-*hard*, there are "good" and "not so good" algorithms for solving it exactly. To the "not so good" algorithms certainly belong most of the simple enumeration methods where one would enumerate all the feasible solutions and then choose the one with the optimal value of the objective function. Such methods have very high time complexity. Among the "good" algorithms belong methods like *branch-and-bound* where an analysis of the problem at hand is used to discard most of the feasible solutions before they are even considered. These approaches allow one to obtain exact solutions of reasonably large problem instances, but their running time still depends exponentially on the size of the problem.
- **Average-case polynomial-time algorithms:** For some problems, it is possible to have algorithms which require superpolynomial-time on only a few instances and for the other instances run in polynomial time. A famous example is the *Simplex Method* for solving problems in Linear Programming.
- **Approximation Algorithms:** We may also relax the requirement of obtaining an exact solution of the optimization problem and content ourselves with a solution which is "not too far" from the optimum. This is partially justified by the fact that, in practice, it is usually enough to obtain a solution that is slightly sub-optimal.

Clearly, there are good approximation algorithms and bad ones as well. What we need is some means of determining the quality of an approximation algorithm and a way of comparing different algorithms. There are a few criteria to consider:

Average-case performance: One has to consider some probability distribution on the set of all possible instances of a given problem. Based on this assumption, an expectation of the performance can then be found. Results of this kind strongly depend on the choice of the initial distribution and do not provide us any information about the performance on a particular instance.

Experimental performance: This approach is based on running the algorithm on a few "typical" instances. It has been used mostly to compare

performance of several approximation algorithms. Of course the result depend on the choice of the "typical" instances and may vary from experiment to experiment.

Worst-case performance: This is usually done by establishing upper and lower bounds for approximate solutions in terms of the optimum value. In case of minimization problems, we try to establish upper bounds, in case of maximization problems, one wants to find lower bounds.

The advantage of the worst-case bounds on the performance of approximation algorithms is the fact that given any instance of the optimization problem, we are guaranteed that the approximate solution stays within these bounds. It should also be noted that approximation algorithms usually output solutions much closer to the optimum than the worst-case bounds suggest. Thus it is of independent interest to see how tight the bounds on the performance of each algorithm are; that is, how bad the approximate solution can really get. This is usually done by providing examples of specific instances for which the approximate solution is very far from the optimum solution.

Establishing worst-case performance bounds for even simple algorithms often requires a very deep understanding of the problem at hand and the use of powerful theoretical results from areas like linear programming, combinatorics, graph theory, probability theory, etc.

We consider an $\mathcal{NP}$-*hard* optimization problem for which as we have seen it is difficult to find the exact optimal solution within polynomial time. At the expense of reducing the quality of the solution by relaxing some of the requirements, we can often get considerable speed-up in the complexity. This leads us to the following definition:

Definition 1. *(Approximation algorithms) Let X be a minimization problem and $\alpha > 1$. An algorithm APP is called an α-approximation algorithm for problem X, if for all instances I of X it delivers in polynomial time a feasible solution with objective value $APP(I)$ such that*

$$APP(I) \leq \alpha OPT(I), \tag{1}$$

where by $APP(I)$ and $OPT(I)$ we denoted the values of an approximate solution and that of an optimal solution for instance I, respectively.

The value α is called the *performance guarantee* or the *worst case ratio* of the approximation algorithm APP. The closer α is to 1 the better the algorithm is.

2. THE GENERALIZED MINIMUM SPANNING TREE PROBLEM

Let $G = (V, E)$ be a complete graph with $V = V_1 \cup \ldots \cup V_m$ partitioned into disjoint clusters V_k, $k \in K = \{1, \ldots, m\}$. Let $c : E \to \mathbb{R}_+$ be a non-negative cost function. The *generalized minimum spanning tree problem* (GMST) is the problem to find a minimum-cost tree spanning a subset of nodes which includes exactly one node from each cluster. We will call a tree containing

one node from each cluster a generalized spanning tree. In [10], Myung *et al.* proved that the GMST problem is $\mathcal{NP}$-hard, and in [13], we presented a stronger result, namely, the GMST problem even on trees is $\mathcal{NP}$-hard as well an exact exponential time algorithm based on dynamic programming.

There are various (slight) generalizations of the GMST problem. For example, the clusters may not be required to be distinct or it may be feasible to chose more than one node per cluster. The latter problem, i.e., the problem of finding a minimum cost tree spanning *at least* one node per cluster, is also known as the *generalized Steiner tree problem*.

Meanwhile, the GMST problem and its relatives have been studied by several authors w.r.t. heuristics, LP-relaxations, polyhedral aspects and approximability, cf., e.g., Dror, Haouari and Chaouachi [2], Dror and Haouari [1], Feremans, Labbe, and Laporte [3], Pop [12, 13], Salazar [14], Garg, Konjevod and Ravi [5]. See also Penn and Rozenfeld [11] for a fairly comprehensive list of existing literature.

In [4], Feremans and Grigoriev proposed an approximation scheme for the GMST problem in a special case of the problem. They considered a geometric case of the problem where the graph has all the vertices situated in the plane and Euclidean distance defines the edge cost.

3. A NEGATIVE RESULT FOR THE GMST PROBLEM

For some hard combinatorial optimization problems it is possible to show that they don't have an approximation algorithm unless $\mathcal{P} = \mathcal{NP}$. In order to give a result of this form it is enough to show that the existence of an α-approximation algorithm would allow one to solve some decision problem, known to be $\mathcal{NP}$-*complete*, in polynomial time.

Applying this scheme to the GMST problem we obtain an in-approximability result. This result is a different formulation in terms of approximation algorithms of a result provided by Myung *et al.* [10] which says that even finding a near optimal solution for the GMST problem is $\mathcal{NP}$-*hard*. Our proof is slightly different to the proof provided in [10].

THEOREM 2. *Under the assumption $\mathcal{P} \neq \mathcal{NP}$, there is no α-approximation algorithm for the GMST problem.*

Proof. Assume that there exists an α-approximation algorithm APP for the GMST problem, where α is a real number greater than or equal to 1. This means that

$$APP(I) \leq \alpha OPT(I),$$

for every instance I, where $OPT(I)$ and $APP(I)$ are the values of the optimal solution and of the solution found by the algorithm APP, respectively.

Then, we will show that APP also solves the node-cover problem for a given graph $G = (V, E)$ and an integer k such that $k < |V|$. This result contradicts the assumption that $P \neq NP$.

We construct a graph $G^{'} = (V^{'}, E^{'})$ and the edge cost function such that the algorithm APP finds a feasible solution with a value no greater than α times the optimal cost if and only if G contains C, where C is a node cover of G, i.e. a subset of V such that all the edges of G are adjacent to at least one node of C. □

The graph $G^{'}$ contains the following $m = |E| + k + 1$ node sets (clusters), $V_1^{'}, ..., V_m^{'}$:

THEOREM 3. *$V_1^{'}$ consists of a single node denoted by r,*

THEOREM 4. *$V_2^{'}, ..., V_{k+1}^{'}$ are identical node sets, each of which has $|V|$ nodes corresponding to the nodes of V, and*

THEOREM 5. *$|E|$ node sets, $V_{k+2}^{'}, ..., V_m^{'}$, each of which contains a single node corresponding to an edge $e \in E$.*

Edges in $G^{'}$ are constructed as follows:

THEOREM 6. *Each node of $V_t^{'}$ for all $t = 2, ..., k+1$ is connected to r by an edge. The set consisting of these edges is denoted by $E_1^{'}$.*

THEOREM 7. *Let i be a node of $V_t^{'}$ for any $t \in \{2, ..., k+1\}$ and j be a node of $V_t^{'}$ for any $t \in \{k+2, ..., m\}$. Then, an edge is constructed between i and j if the edge of G corresponding to j is incident to the node of G corresponding to i, and let $E_2^{'}$ denote the set of those edges.*

THEOREM 8. *We also construct an edge between i and j even though the edge of G corresponding to j is not incident to the node of G corresponding to i, and we let $E_3^{'}$ denote the set of those edges.*

Proof. We let $E^{'} = E_1^{'} \cup E_2^{'} \cup E_3^{'}$.
The cost of each edge is defined as follows:

$$c_{ij} = \begin{cases} 0 & \text{for all } i, j \in E_1^{'} \\ 1 & \text{for all } i, j \in E_2^{'} \\ (|E|+1)\alpha & \text{for all } i, j \in E_3^{'}. \end{cases}$$

We claim that G contains C if and only if

$$APP(I) \leq \alpha |E|,$$

where instance I corresponds to $G^{'}$ and its cost function.

Note that there always exists a generalized spanning tree in $G^{'}$: all the clusters different from the identical clusters $V_2^{'}, ..., V_{k+1}^{'}$ have only one node and if we select k nodes from $V_2^{'}, ..., V_{k+1}^{'}$, one node from each cluster such that each node of C is included, then these k nodes together with the remaining nodes selected always uniquely form a generalized spanning tree of $G^{'}$ using edges in $E_1^{'} \cup E_2^{'}$, by the definition of $G^{'}$.

Suppose now that $G^{'}$ contains a generalized spanning tree and let C be a set of distinct nodes selected from the clusters, $V_2^{'}, ..., V_{k+1}^{'}$ in the tree, then C is a node cover of G.

Therefore, we showed that a generalized spanning tree only using edges in $E_1^{'} \cup E_2^{'}$ exists in $G^{'}$ if and only if G contains a node cover C.

If G contains C, then $OPT(I) = |E|$. Moreover, if G does not contain C, then any generalized spanning tree of $G^{'}$ should use at least one edge in $E_3^{'}$ and thus

$$OPT(I) \geq |E| - 1 + \alpha(|E| + 1) > \alpha|E|.$$

In particular if G contains a node cover C then the approximation algorithm APP will produce a solution with value

$$APP(I) \leq \alpha|E| = \alpha OPT(I),$$

i.e. the solution does not use any edge from $E_3^{'}$ and APP identifies a node cover. □

However, under further assumptions, in the next section we present a positive result for the GMST problem.

4. AN APPROXIMATION ALGORITHM FOR THE GMST PROBLEM

In this section we provide a polynomial time approximation algorithm for the *metric* GMST with *bounded cluster size*. I.e., we consider the case where the cost function $c : E \to \mathbb{R}_+$ satisfies the triangle inequality

$$c_{ij} \leq c_{ik} + c_{kj} \qquad (i, j, k \in V)$$

and the clusters are bounded by

$$|V_k| \leq \rho, \qquad k \in K$$

for some $\rho > 0$. For this class of problem instances we can efficiently construct a solution with cost at most 2ρ times the optimum (which is admittedly rather poor for practical purposes).

Our approach is based on ideas from Slavik [16] where a similar type of approximation algorithms is described for the so-called *generalized* TSP, the problem of determining a shortest cycle through m nodes, one from each cluster V_k. The additional difficulty we encounter in our situation (as compared to the general TSP) is to compare the length of a minimum cost spanning tree to the minimum of $c^{\mathrm{T}}x$ over the cut polytope (or subtour elimination polytope in the TSP terminology), cf. Lemma 3 in Section 5.

We consider the GMST problem on the graph G=(V,E) and assume that the cluster size is bounded, $|V_k| \leq \rho,\ k \in K$ and that the cost function c satisfies the triangle inequality. We introduce the binary variables $x \in \mathbb{R}^E$ and $y \in \mathbb{R}^V$ with values $x_e = 1$ if edge e is selected, $x_e = 0$ otherwise and $y_i = 1$ if node i is selected, $y_i = 0$ otherwise.

The optimum value of the corresponding GMST can then be defined by the following integer linear program

$$\begin{array}{lllll} \text{(ILP)} & z^*_{\text{ILP}} & = & \min\ c^{\mathrm{T}}x & \\ & & & x(\delta(S)) \geq y_i, & V_1 \subseteq S \subseteq V\backslash\{i\} \\ & & & y(V_k) = 1, & k \in K \\ & & & x \in \{0,1\}^E,\ y \in \{0,1\}^V. & \end{array}$$

Here, as usual, $\delta(S) \subseteq E$ denotes the *cut* induced by $S \subseteq V$, i.e., the set of edges joining S to its complement $V\backslash S$. Furthermore, we use the general shorthand notation

$$x(F) = \sum_{f \in F} x_f \quad \text{resp.} \quad y(U) = \sum_{u \in U} y_u$$

for $F \subseteq E$ resp. $U \subseteq V$. Thus $y(V_k) = 1$ means that exactly one node is "picked" in each cluster. Furthermore, the *cut constraints* $x(\delta(S)) \geq y_i$ ensure that $x \in \{0,1\}^E$ connects all selected nodes (to the node v_1 selected in the "root cluster" V_1).

Obviously the GMST's of the instance defined by G, $V = V_1 \cup \ldots \cup V_m$ and $c \in \mathbb{R}^E_+$ are in $1-1$ correspondence with optimum solutions of (ILP).

For each possible *root* $v \in V_1$, we consider the corresponding *rooted relaxation* of (ILP)

$$\begin{array}{lllll} \text{(LP}(v)\text{)} & z^*(v) & = & \min\ c^T x & \\ & & & x(\delta(S)) \geq y_i & , \quad v \in S \subseteq V\backslash\{i\} \\ & & & y(V_k) = 1 & , \quad k \in K \\ & & & y(v) = 1 & \\ & & & x, y \geq 0. & \end{array}$$

By assumption, $|V_1| \leq \rho$, so there are at most ρ LP's that we solve. Each of these can be solved in polynomial time (relative to the input size of (GMST)) by means of the Ellipsoid Method (cf., e.g., [8]). (Note that we can check whether a given (x, y) is feasible by computing a maximum $v - i$ flow w.r.t. edge capacities $x \in \mathbb{R}^E_+$ for all $i \in V$). Furthermore, clearly,

$$\min_{v \in V_1} z^*(v) \leq z^*_{\text{ILP}} \tag{2}$$

holds.

Assume that $v_1 \in V_1$ achieves the minimum in (2) and let (x^*, y^*) be an optimal solution of (LP(v_1)). Since $|V_k| \leq \rho$ holds by assumption, we may choose in each cluster V_k a node $v_k \in V_k$ with

$$y^*(v_k) \geq 1/\rho. \tag{3}$$

□

Let $W = \{v_1, \ldots, v_m\} \subseteq V$ denote the resulting set of chosen nodes. We now compute an MST on W (relative to the given cost function c) and claim that this tree, say $T = T(W)$, has cost $c(T)$ at most 2ρ times the optimum z^*_{ILP}. More precisely, we show that

THEOREM 9. *The tree $T = T(W)$ has cost at most $2\rho\, z^*(v_1)$.*

5. PROOF OF CORRECTNESS

Our crucial argument relies on the following result of [6] (extending earlier work of [9]): Consider fixed *connectivity requirements* $r_{ij} \geq 0$ $(i, j \in V)$ that are *symmetric* in the sense that $r_{ij} = r_{ji}$ for all $i, j \in V$. Then, for $D \subseteq V$, the linear programs

$$\begin{array}{lll} (R_D) & \min\ c^{\mathrm{T}}x & \\ & x(\delta(S)) \geq r_{ij} & , \quad i \in S \subseteq V\backslash\{j\} \\ & x(\delta(\{i\})) = \max\limits_{j \neq i}\ r_{ij} & , \quad i \in D \\ & x \geq 0 & \end{array}$$

have all the same optimum value (independent of $D \subseteq V$). This is referred to as the *parsimonious property* (as we can force the least possible value of x on each elementary cut $\delta(\{i\})$).

We use this result as follows. Let $W = \{v_1, \ldots, v_m\} \subseteq V$ be the set of chosen nodes (cf. Section 2). Define

$$r_{ij} := \begin{cases} 1/\rho & i, j \in W \\ 0 & \text{else.} \end{cases} \tag{4}$$

Then the optimum solution (x^*, y^*) of LP(v_1) is a feasible solution of $(R_\emptyset)$ (relative to r_{ij} as in (4)): Indeed, let $i, j \in V$ and $i \in S \subseteq V\backslash\{j\}$. If either $i \notin W$ or $j \notin W$, then $r_{ij} = 0$ and $x^*(\delta(S)) \geq 0 = r_{ij}$ holds trivially (since $x^* \geq 0$). Thus assume $i, j \in W$, so $r_{ij} = 1/\rho$. Furthermore, assume w.l.o.g. that $v_1 \in S$ (otherwise, replace S by $V\backslash S$). The feasibility of (x^*, y^*) then implies

$$x^*(\delta(S)) \geq y^*_j \geq 1/\rho = r_{ij},$$

as claimed.

Thus we conclude that (relative to r_{ij} as in (4)) the optimum value z^* of $(R_\emptyset)$ –and hence of (R_V)– is less than or equal to $z^*(v_1)$:

$$\begin{array}{lll} z^*(v_1) \geq \overline{z} \ = \ \min\ c^{\mathrm{T}}x & & \\ \qquad x(\delta(S)) \geq 1/\rho & i \in S \subseteq V\backslash\{j\}, \quad i, j \in W \\ \qquad x(\delta(\{i\})) = 0 & i \in V\backslash W \\ \qquad x(\delta(\{i\})) = 1/\rho & i \in W \\ \qquad x \geq 0. & \end{array} \tag{5}$$

To prove Theorem 2, we are left to show that a min cost tree T in the (complete) subgraph induced by W has cost $c(T) \leq 2\rho\, \overline{z}$.

LEMMA 10. *A min cost tree T spanning all nodes in W has cost $c(T) \leq 2\rho\, \overline{z}$.*

Proof. The value of a min cost spanning tree in the (complete) subgraph $(W, E(W))$ induced by W is given by (cf. eg. Schrijver [15, (50.12)])

$$\begin{array}{lllll} \text{(MST)} & \min\ c^{\mathrm{T}}x & & & \\ & x(E(S)) & \leq & |S|-1, & \emptyset \neq S \subseteq W \\ & x(E(W)) & = & |W|-1, & \\ & x \geq 0 & & & \end{array}$$

Here, $E(W)$ is the set of edges induced by W and $E(S)$ is the set of edges induced by $S \subseteq W$. Furthermore, by slightly misusing the notation, the vectors x and c in (MST) are restricted to the edges in $E(W)$.

To prove the claim of the lemma, it suffices to show that a feasible (optimal) solution $\overline{x} \in \mathbb{R}^E$ of the LP in (5) (which obviously has its support contained in $E(W)$) gives rise to a feasible solution

$$\widetilde{x} := 2\rho(1 - \tfrac{1}{|W|})\, \overline{x}_{|_{E(W)}} \in \mathbb{R}^{E(W)}$$

of (MST), implying that the optimal tree cost is

$$c(T) \leq 2\rho(1 - 1/|W|)\, \overline{z} \leq\ 2\rho\, \overline{z}\ .$$

First note that the equality constraints in (5) imply

$$\overline{x}(E) = \overline{x}(E(W)) = \tfrac{1}{2} \sum_{i \in W} \overline{x}(\delta(\{i\})) = \tfrac{1}{2\rho}|W|\ ,$$

so

$$\widetilde{x}(E(W)) = (1 - \tfrac{1}{|W|})|W| = |W| - 1\ ,$$

as required. Furthermore, if $S \subseteq W$, adding the constraints $\overline{x}(\delta(\{i\})) = 1/\rho$ for $i \in S$, we get

$$2\overline{x}(E(S)) + \overline{x}(\delta(S)) = \tfrac{|S|}{\rho}\ .$$

Hence $\overline{x}(\delta(S)) \geq 1/\rho$ yields $\overline{x}(E(S)) \leq \frac{1}{2}(|S| - 1)/\rho$ and consequently,

$$\widetilde{x}(E(S)) \leq (1 - \tfrac{1}{|W|})(|S| - 1) \leq |S| - 1\ ,$$

and the claim follows. □

We finally comment on the variants of GMST were the clusters are non-disjoint or where at least one node has to be selected per cluster. Our approximation algorithm applies to these variants as well. Indeed, for the first case no change is necessary. For the second variant, one has to solve $O(2^\rho)$ rooted relaxations (one for each possible choice of roots $W_1 \subseteq V_1$) with cut constraints

$$x(\delta(S))) \geq y_i, \qquad v \in S \subseteq V \backslash \{i\},\ v \in W_1$$

to ensure connectivity of $\mathrm{supp}(x) \subseteq E$. All other arguments remain unchanged.

REFERENCES

[1] Dror, M. and Haouari, M., *Generalized Steiner problem and other variants*, J. Combinatorial Optimization, **4**, pp. 415–436, 2000.

[2] Dror, M., Haouari, M. and CHAOUACHI J., *Generalized spanning trees*, EJOR, **120**, pp. 583–592, 2000.

[3] Feremans, C., Labbe, M. and LAPORTE, G., *A comparative analysis of several formulations for the generalized minimum spanning tree problem*, Networks, **39 (1)**, pp. 29–34, (2002).

[4] Feremans, C. and Grigoriev, A., *An Approximation Scheme for the Generalized Geometric Minimum Spanning Tree Problem with Grid Clustering*, preprint, 2004.

[5] Garg, N., Konjevod, G. and Ravi, R., *A polylogarithmic algorithm for the group Steiner tree problem*, J. of Algorithms, **37 (1)**, pp. 66–84, 2000.

[6] Goemans, M.X and Bertsimas, D.J., *Survivable networks, linear programming relaxations and parsimonious property*, Mathematical programming, **60**, pp. 145–166, 1993.

[7] Grötschel, M., *Discrete mathematics in manufacturing*, Preprint SC 92–3, ZIB, 1992.

[8] Grötschel, M., Lovász, L. and Schrijver, A., *Geometric Algorithm and Combinatorial Optimization*, Springer Verlag, Berlin, 1988.

[9] Lovász, L., *On some connectivity properties of Eulerian graphs*, Acta Mathematica Acad. Scient. Hungaricae, **28**, pp. 129–138, 1976.

[10] Myung, Y.S., Lee, C.H. and Tcha, D.W., *On the Generalized Minimum Spanning Tree Problem*, Networks, **26**, pp. 513–623, 1995.

[11] Penn, M. and Rozenfeld S., *Approximation algorithm for the group Steiner network problem*, Technical report, Faculty of Industrial Engineering, Haifa, Israel, Oct. 2003.

[12] Pop, P.C., *The Generalized Minimum Spanning Tree Problem*, PhD thesis, University of Twente, The Netherlands, 2002.

[13] Pop, P.C., *New Models of the Generalized Minimum Spanning Tree Problem*, Journal of Mathematical Modelling and Algorithms, **3**, no. 2, pp. 153–166, 2004.

[14] Salazar, J.J., *A note on the generalized Steiner tree polytope*, Discrete Appl. Math. **100**, nos. 1–2, pp. 137–144, 2000.

[15] Schrijver, A., *Combinatorial Optimization*, Springer-Verlag, Berlin, 2003.

[16] Slavik, P., *On the approximation of the generalized Traveling salesman problem*, preprint, 1999.

Received by the editors: November 12, 2004.