Available online at www.sciencedirect.com
SCIENCE DIRECT®

European Journal of Operational Research 170 (2006) 900–908

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Discrete Optimization

A new relaxation method for the generalized minimum spanning tree problem

Petrica C. Pop [a,*], W. Kern [b], G. Still [b]

[a] Department of Mathematics and Computer Science, Faculty of Sciences, North University of Baia Mare, Baia Mare 430406, Romania
[b] Faculty of Mathematical Sciences, University of Twente, P.O. Box 217, 7500 Enschede, The Netherlands

Received 7 March 2003; accepted 28 July 2004
Available online 12 October 2004

Abstract

We consider a generalization of the minimum spanning tree problem, called the generalized minimum spanning tree problem, denoted by GMST. It is known that the GMST problem is $\mathcal{NP}$-hard. We present several mixed integer programming formulations of the problem. Based on a new formulation of the problem we give a new solution procedure that finds the optimal solution of the GMST problem for graphs with nodes up to 240. We discuss the advantages of our approach in comparison with earlier methods.

Keywords: Combinatorial optimization; Minimum spanning trees; Generalized minimum spanning tree problem

1. Introduction

We consider the generalized version of the minimum spanning tree problem (MST) called the generalized minimum spanning tree problem (GMST). Given an undirected graph whose nodes are partitioned into a number of subsets (clusters), the GMST problem is then to find a minimum-cost tree which includes *exactly* one node from each cluster. Therefore, the MST is a special case of the GMST problem where each cluster consists of exactly one node.

The GMST problem has been introduced by Myung, Lee and Tcha in [8] and the same authors showed that the problem is $\mathcal{NP}$*-hard*. A stronger result regarding its complexity has been provided by Pop [9]

* Corresponding author.
E-mail address: pop_petrica@yahoo.com (P.C. Pop).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2004.07.058

namely, the GMST problem even defined on trees is *$\mathcal{NP}$-hard*. The GMST problem has several applications to location and telecommunications problems, see [7] and [10].

Myung et al. [8] used a branch and bound procedure in order to solve the GMST problem. Their lower procedure is a heuristic method which approximates the linear programming relaxation associated with the dual of the multicommodity flow formulation of the GMST problem. They developed also a heuristic algorithm which finds a primal feasible solution for the GMST problem using the obtained dual solution. The GMST problem was solved to optimality for nodes up to 200 by Feremans [3] using a branch-and-cut algorithm. More information on the problem can be found in [3,4,8,9].

A variant of the GMST problem is the problem of finding a minimum cost tree including *at least* one vertex from each cluster. This problem was introduced by Dror et al. in [2]. These authors provide also five heuristics including a genetic algorithm. In the present paper we confine ourselves to the problem of choosing exactly one vertex per cluster.

Related work is to be found in [1] where Dror and Haouari present the generalized version of several combinatorial optimization problems including the generalized traveling salesman problem, the generalized Steiner tree problem, the generalized assignment problem, etc.

2. Definition and complexity of the GMST problem

Let $G = (V, E)$ be an n-node undirected graph. Let $V_1, \ldots, V_m$ be a partition of V into m subsets called *clusters* (i.e., $V = V_1 \cup V_2 \cup \ldots \cup V_m$ and $V_l \cap V_k = \emptyset$ for all $l, k \in \{1, \ldots, m\}$ with $l \neq k$) and denote by $K = \{1, \ldots, m\}$ the index of the clusters. We assume that that edges are defined only between nodes which belong to different clusters and we denote the cost of an edge $e = (i, j) \in E$ by c_{ij} or by $c(i, j)$.

The *generalized minimum spanning tree* (GMST) problem asks for finding a minimum-cost tree T spanning a subset of nodes which includes exactly one node from each cluster V_i, $i \in \{1, \ldots, m\}$. We will call such a tree a *generalized spanning tree*.

In [8], Myung et al. proved that the GMST problem is $\mathcal{NP}$-hard. We proved in [9] a stronger result:

Theorem 1. *The Generalized Minimum Spanning Tree problem on trees is $\mathcal{NP}$-hard.*

The proof of this result is based on a polynomially reduction of the set cover problem, which is known to be $\mathcal{NP}$-hard (see for example [6]), to the GMST problem defined on trees.

3. Integer programming formulations

The GMST problem can be formulated as an integer program in many different ways, cf. [3,4,8], and [9]. For example, introducing the variables $x_e \in \{0,1\}$, $e \in E$ and $z_i \in \{0,1\}$, $i \in V$, to indicate whether an edge e respectively a node i is contained in the spanning tree, we obtain a valid formulation (so-called *generalized cutset formulation*, introduced in [8]) as follows:

$$\min \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad z(V_k) = 1 \quad \forall k \in \{1, \ldots, m\}, \tag{1}$$

$$x(\delta(S)) \geqslant z_i + z_j - 1 \quad \forall i \in S \subset V, \ \ j \notin S, \tag{2}$$

$$x(E) = m - 1, \tag{3}$$

$$x_e \in \{0, 1\} \quad \forall e \in E, \tag{4}$$

$$z_i \in \{0, 1\} \quad \forall i \in V. \tag{5}$$

Here we use the standard shorthand notations:

$$x(F) = \sum_{e \in F} x_e, \quad F \subseteq E, \quad \text{and} \quad z(S) = \sum_{i \in S} z_i, \quad S \subseteq V,$$

and for $S \subseteq V$, the cutset, denoted by $\delta(S)$, is defined as usually:

$$\delta(S) = \{e = (i,j) \in E \mid i \in S, j \notin S\}.$$

The constraints in the generalized cutset formulation imply that a feasible solution defines a connected subgraph (constraints (2)), with $m-1$ edges (constraint (3)) and exactly one node from each cluster (constraints (1)), i.e. a generalized spanning tree.

The generalized cutset formulation has exponentially many constraints since we have to choose subsets S of V (constraints (2)). We consider in this paper computational approaches based on models with a polynomial number of constraints.

The approach in [8] (to which we will compare our own results later) is based on the so-called multicommodity flow model.

The idea is to consider a generalized spanning tree T as a directed tree, rooted at some node V_1. In this model every $k \in K \setminus \{1\}$ defines a commodity and one unit of flow of some commodity k originates from V_1 and must be delivered to V_k ($k = 2, \dots, m$) along T. Formally, we let A denote the set containing two oppositely directed arcs for every $e \in E$. Furthermore, we introduce capacity variables $w \in \mathbb{R}^A$ and flow variables $f^k \in \mathbb{R}^A$ (indicating the amount of flow $f_a^k \leqslant w_a$ of commodity k on arc a). With symmetric arc costs $c_{ij} = c_{ji}$, the model can be written as

$$\begin{aligned}
\min \quad & \sum_{a \in A} c_a w_a \\
\text{s.t.} \quad & z(V_k) = 1 \quad \forall k \in K = \{1, \dots, m\}, \\
& w(A) = m - 1, \\
& \sum_{a \in \delta^+(i)} f_a^k - \sum_{a \in \delta^-(i)} f_a^k = \begin{cases} z_i, & i \in V_1, \\ -z_i, & i \in V_k, \\ 0, & i \notin V_1 \cup V_k, \end{cases} \qquad k \in K_1, \\
& f_{ij}^k \leqslant w_{ij} \quad \forall a = (i,j) \in A, \ k \in K_1, \\
& w_{ij} + w_{ji} = x_e \quad \forall e = (i,j) \in E, \\
& f_a^k \geqslant 0 \quad \forall a = (i,j) \in A, \ k \in K_1, \\
& x, z \in \{0, 1\}.
\end{aligned}$$

The computational approach in Myung et al. [8] is to solve the linear programming relaxation of the above formulation and use the resulting lower bound in a branch and bound method. (More precisely, they compute only approximately the optimum value, using a dual ascent method.)

Let G' be the graph obtained from G after replacing all nodes of a cluster V_i with a supernode representing V_i. For convenience, we identify V_i with the supernode representing it. We will call this graph the global graph. We assume that G' with vertex set $\{V_1, \dots, V_m\}$ is complete.

Our last model arises from distinguishing between *global* variables, i.e. variables modelling the inter-cluster (global) connections, and *local* ones, i.e. expressing whether an edge is selected between two clusters linked in the global graph. We introduce variables y_{ij} ($i, j \in \{1, \dots, m\}$) to describe the global connections. So $y_{ij} = 1$ if cluster V_i is connected to cluster V_j and $y_{ij} = 0$ otherwise and we assume that y represents a spanning tree. The convex hull of all these y-vectors is generally known as the spanning tree polytope (on the global graph G' which we assumed to be complete).

Following Yannakakis [11] this polytope, denoted by P_{MST}, can be represented by the following polynomial number of constraints:

$$\sum_{\{i,j\}} y_{ij} = m-1,$$

$$y_{ij} = \lambda_{kij} + \lambda_{kji} \quad \text{for } 1 \leqslant k,i,j \leqslant m \text{ and } i \neq j, \tag{6}$$

$$\sum_{j} \lambda_{kij} = 1 \quad \text{for } 1 \leqslant k,i,j \leqslant m \text{ and } i \neq k, \tag{7}$$

$$\lambda_{kkj} = 0 \quad \text{for } 1 \leqslant k,j \leqslant m, \tag{8}$$

$$y_{ij}, \lambda_{kij} \geqslant 0 \quad \text{for } 1 \leqslant k,i,j \leqslant m,$$

where the variables λ_{kij} are defined for every triple of nodes k, i, j, with $i \neq j \neq k$ and their value for a spanning tree is

$$\lambda_{kij} = \begin{cases} 1, & \text{if } j \text{ is the parent of } i \text{ when we root the tree at } k, \\ 0, & \text{otherwise.} \end{cases}$$

The constraints (6) mean that an edge (i,j) is in the spanning tree if and only if either i is the parent of j or j is the parent of i; the constraints (7) mean that if we root a spanning tree at k then every node other than node k has a parent and finally constraints (8) mean that the root k has no parent.

If the vector y describes a spanning tree on the global graph G', which we shall refer as the global spanning tree, then the corresponding best (w.r.t. minimization of the costs) generalized spanning tree can be obtained either by using dynamic programming, see [9], or by solving the following 0–1 programming problem:

$$\begin{aligned} \min \ & \sum_{e \in E} c_e x_e \\ \text{s.t. } & z(V_k) = 1 \quad \forall k \in K = \{1,\ldots,m\}, \\ & x(V_l, V_r) = y_{lr} \quad \forall l,r \in K = \{1,\ldots,m\}, l \neq r, \\ & x(i, V_r) \leqslant z_i \quad \forall r \in K, \forall i \in V \setminus V_r, \\ & x_e, z_i \in \{0,1\} \quad \forall e = (i,j) \in E, \forall i \in V, \end{aligned}$$

where $x(V_l, V_r) = \sum_{i \in V_l, j \in V_r} x_{ij}$ and $x(i, V_r) = \sum_{j \in V_r} x_{ij}$.

For given y, we denote the feasible set of the linear programming relaxation of this program by $P_{local}(y)$. The following result holds:

Proposition 2. *If y is the 0–1 incidence vector of a spanning tree of the contracted graph then the polyhedron $P_{local}(y)$ is integral.*

Proof. Suppose that the 0–1 vector y describes a spanning tree T of the contracted graph G', then in order to prove that the polyhedron $P_{local}(y)$ is integral it is enough to show that every solution of the linear programming relaxation can be written as a convex combination of solutions corresponding to spanning trees.

To prove the above assertion we use backward induction on $|supp(x)|$, where by $supp(x)$ we denoted the support of the vector of solutions x, which is defined as follows:

$$supp(x) := \{e | x_e \neq 0, \ e \in E\}.$$

Suppose that there is a global connection between the clusters V_l and V_r (i.e. $y_{lr} = 1$) then

$$1 = x(V_l, V_r) = \sum_{i \in V_l} x(i, V_r) \leqslant \sum_{i \in V_l} z_i = 1,$$

which implies that $x(i, V_r) = z_i$.

We claim that $supp(x) \subseteq E$ contains a tree connecting all clusters. This implies that the initial step is true and also helps us in proving the induction step.

Assume the contrary and let $T^1 \subseteq E$ be a maximal tree in $supp(x)$. Since T^1 does not connect all clusters, there is some edge (l, r) with $y_{lr} = 1$ such that T^1 has some vertex $i \in V_l$ but no vertex in V_r. Then $z_i > 0$, and thus $x(i, V_r) = z_i > 0$, so T^1 can be extended by some $e = (i, j)$ with $j \in V_r$, a contradiction.

We assume that a solution x of the linear programming relaxation, having the support $supp(x)$ can be written as a convex combination of solutions corresponding to trees and we will prove that a solution $\widehat{x}$ of the linear programming relaxation, having the support $|supp(\widehat{x})| = |supp(x)| - 1$ can be written as a convex combination of solutions corresponding to trees.

Now let x^{T^1} be the incidence vector of T^1 and let

$$\alpha := \min\{x_e \mid e \in T^1\}.$$

If $\alpha = 1$, then $x = x^{T^1}$ and we are done.

Otherwise, let z^{T^1} be the vector which has $z_i^{T^1} = 1$ if T^1 covers $i \in V$ and $z_i^{T^1} = 0$ otherwise. Then

$$(\widehat{x}, \widehat{z}) := ((1-\alpha)^{-1}(x - \alpha x^{T^1}), (1-\alpha)^{-1}(z - \alpha z^{T^1}))$$

is again in $P_{local}(y)$ and, by induction, it can be written as a convex combination of tree solutions. The claim follows. $\square$

A similar argument shows that the polyhedron $P_{local}(y)$ is integral even in the case when the 0–1 vector y describes a cycle free subgraph in the contracted graph. If the 0–1 vector y contains a cycle of the contracted graph then $P_{local}(y)$ is in general not integral. In order to show this we consider the following example:

If the lines drawn in Fig. 1 (i.e., {1,3}, {2,4} etc.) have cost 1 and all the other lines (i.e., {1,4}, {2,3} etc.) have cost $M \gg 1$, then $z \equiv \frac{1}{2}$ and $x \equiv \frac{1}{2}$ on the drawn lines is an optimal solution of $P_{local}(y)$, showing that the polyhedron $P_{local}(y)$ is not integral.

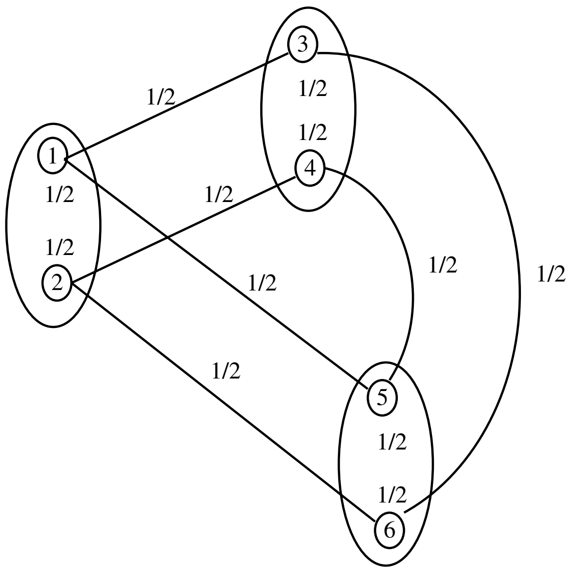

Fig. 1. Example showing that $P_{local}(y)$ may have fractional extreme points.

The observations presented so far lead to our final formulation, called *local-global formulation* of the GMST problem as an 0–1 mixed integer programming problem, where only the global variables y are forced to be integral:

$$\begin{aligned} (P)\ \ \min\ & \sum_{e\in E} c_e x_e, \\ \text{s.t.}\ & y \in P_{MST}, \\ & (x,z) \in P_{local}(y), \\ & y_{lr} \in \{0,1\} \quad \forall 1 \leqslant l, r \leqslant m. \end{aligned}$$

This new formulation of the GMST problem was obtained by incorporating the constraints characterizing P_{MST}, with $y \in \{0,1\}$, into $P_{local}(y)$.

In the next section we present a solution procedure for solving the GMST problem based on the local-global formulation and we report on our computational results for many instances of the problem.

4. A new solution procedure and computational results

There are different ways to solve the GMST problem with the help of formulation (P). The first possibility is to consider the mixed integer program (P) and solve it directly (for example with CPLEX).

Secondly, if for (P) we consider the constraints characterizing P_{MST} only for fixed k, $1 \leqslant k \leqslant m$, then we get a relaxation, denoted by P^k, of P. Using the the description of Yannakakis for the global spanning tree polytope, this situation corresponds to the case when we choose randomly one cluster V_k and root the global tree only at the root k.

$$\begin{aligned} (P^k)\ \ \min\ & \sum_{e\in E} c_e x_e \\ \text{s.t.}\ & z(V_k) = 1 \quad \forall k \in K = \{1,\ldots,m\}, \\ & x(E) = m-1, \\ & x(V_l, V_r) = y_{lr} \quad \forall l, r \in K = \{1,\ldots,m\}, l \neq r, \\ & x(i, V_r) \leqslant z_i \quad \forall r \in K, \forall i \in V \setminus V_r, \\ & y_{ij} = \lambda_{kij} + \lambda_{kji} \quad \forall 1 \leqslant k, i, j \leqslant m \text{ and } i \neq j, k \text{ fixed}, \\ & \sum_j \lambda_{kij} = 1 \quad \forall 1 \leqslant k, i, j \leqslant m \text{ and } i \neq k, k \text{ fixed}, \\ & \lambda_{kkj} = 0 \quad \forall 1 \leqslant k, j \leqslant m, k \text{ fixed}, \\ & \lambda_{kij} \geqslant 0 \quad \forall 1 \leqslant k, i, j \leqslant m, k \text{ fixed}, \\ & x_e, z_i \geqslant 0 \quad \forall e = (i,j) \in E, \forall i \in V, \\ & y_{lr} \in \{0,1\} \quad \forall 1 \leqslant l, r \leqslant m. \end{aligned}$$

If the optimal solution of this relaxation (solved with CPLEX) produces a generalized spanning tree, then we have given the optimal solution of the GMST problem. Otherwise we get a subgraph containing at least one cycle and we add the corresponding constraints (from the characterization of P_{MST}) in order to break that cycle (i.e. root the global tree also in a second cluster, contained in the cycle) and proceed in this way till we get the optimal solution of the GMST problem. We call this procedure the *rooting procedure*.

It turned out that the lower bounds computed by solving the linear programming relaxation P^k are comparable with the lower bounds provided in [8], but can be computed faster.

Our algorithms have been coded in C and compiled with a HP-UX cc compiler. For solving the linear and mixed integer programming problems we used CPLEX 6.5. The computational experiments were performed on a HP 9000/735 computer with a 125 Mhz processor and 144 Mb memory.

According to the method of generating the edge costs, the problems generated are classified into two types: the Euclidean case and the non-Euclidian case.

The clusters in both cases are random and we assume that every cluster has the same number of nodes.

In the non-Euclidean model the edge costs are randomly generated on [0, 100]. For each type of instance we considered five trials. We compare the computational results in this case, obtained for solving the problem using our rooting procedure with the computational results given by Myung et al. in [8] and Feremans in [3]. The computational results are presented in Table 1.

In the Euclidean case we used the grid clustering described in Fischetti et al. [5]. The cost between nodes are the Euclidean distances between the nodes. In this case, the clusters can be interpreted as physical clusters and models the geographical applications (e.g. cities corresponding to nodes and clusters corresponding to countries, regions or counties). In the other model such an interpretation is not valid. The computational

Table 1
Computational results for non-Euclidean problems (average of five trials per type of instance)

Pb. size		Rooting procedure		Branch and cut [3]		Myung's results	
m	n	LB/OPT	CPU	LB/UB	CPU	LB/OPT	CPU
8	24	100	0.0	100	0.0	100	0.0
	32	100	0.0	100	0.2	100	0.2
	48	100	0.2	100	1.4	94.3	3.2
	80	100	0.6	100	4.2	94.9	17.6
10	30	100	0.1	100	1.0	89.1	0.0
	40	100	0.7	100	1.0	–	–
	60	100	0.9	100	3.2	87.8	3.2
	100	100	3.5	100	8.8	91.3	17.6
12	36	100	0.1	100	1.8	89.6	6.0
	48	100	1.6	99.2	3.2	91.3	54.9
	72	100	5.6	100	6.8	100	6.8
	120	100	14.5	–	–	–	–
15	45	100	0.2	100	3.6	89.0	17.4
	90	100	5.9	100	21.4	–	–
	150	100	40.5	98.8	42.4	–	–
18	54	100	0.5	99.5	7.6	–	–
	108	100	9.4	–	–	–	–
	180	100	193.8	–	–	–	–
20	60	100	3.8	–	–	–	–
	120	100	11.4	96.3	39.8	–	–
	200	100	407.6	94.6	191.4	–	–
25	75	100	21.6	–	–	–	–
	150	100	25.1	88.3	178.8	–	–
	200	100	306.6	97.8	140.6	–	–
30	90	100	40.0	–	–	–	–
	180	100	84.0	96.6	114.6	–	–
	240	100	341.1	–	–	–	–
40	120	100	71.6	100	92.6	–	–
	160	100	1713.2	94.2	288.6	–	–

Table 2
Computational results for Euclidean problems (average of five trials per type of instance)

Pb. size		Rooting procedure		
m	n	LB/OPT	CPU	Number of roots
8	24	100	0.1	1
	48	100	0.9	2
	80	100	26.5	2
10	30	100	0.3	2
	60	100	6.6	3
	100	100	45.2	3
12	36	100	0.4	2
	72	100	57.6	3
	120	100	94.2	3
15	45	100	3.2	3
	90	100	236.9	3
	150	100	423.5	4
18	54	100	20.2	4
	108	100	363.6	4
20	60	100	43.8	4
	160	100	869.8	4
30	120	100	74.0	4
	150	100	856.8	5
40	120	100	101.5	5

results obtained for solving the GMST problem in this case with the rooting procedure are presented in Table 2.

The first two columns in the tables give the size of the problem: the number of clusters (m) and the number of nodes (n). The next columns describe the rooting procedure and contain: the lower bounds obtained as a percentage of the optimal value of the GMST problem (LB/OPT) and the computational times (CPU) in seconds for solving the GMST problem and in addition in the second table the minimum number of roots chosen by the rooting procedure in order to get the optimal solution of the GMST problem. The last columns in the first table contain the lower bounds as a percentage of the upper bounds of the GMST problem (LB/UB) and the computational times (CPU) in seconds for solving the GMST problem with the branch and cut algorithm [3] and the lower bounds as a percentage of the optimal value of the GMST problem (LB/OPT) and the computational times (CPU) obtained by Myung [8]. In the table the sign '–' means that the corresponding information was not provided in [3] or [8].

As it can be seen, in all the instances that we considered, for graphs with nodes up to 240, the optimal solution of the GMST problem has been found by using our rooting procedure. It is worth to mention that for the instances considered in the table, the maximum number of clusters chosen as roots, in order to get the optimal solution of the problem, was 5. These numerical experiences with the new formulation of the GMST problem are very promising.

Acknowledgment

We are grateful to M. Labbé for pointing out a mistake in an earlier version of this paper.

References

[1] M. Dror, M. Haouari, Generalized Steiner problems and other variants, Journal of Combinatorial Optimization 4 (2000) 415–436.
[2] M. Dror, M. Haouari, J. Chaouachi, Generalized spanning trees, European Journal of Operational Research 120 (2000) 583–592.
[3] C. Feremans, Generalized Spanning Trees and Extensions, PhD thesis, Universite Libré de Bruxelles, Belgium, 2001.
[4] C. Feremans, M. Labbé, G. Laporte, A comparative analysis of several formulations for the generalized minimum spanning tree problem, Networks 39 (1) (2002) 29–34.
[5] M. Fischetti, J.J. Salazar, P. Toth, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem, Operations Research 45 (1997) 378–394.
[6] M.R. Garey, D.S. Johnson, Computers and Intractability: A guide to the theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
[7] M. Gerla, L. Frata, Tree structured fiber optics MAN's, IEEE Journal on Selected Areas in Communication SAC-6 (1988) 934–943.
[8] Y.S. Myung, C.H. Lee, D.W. Tcha, On the generalized minimum spanning tree problem, Networks 26 (1995) 513–623.
[9] P.C. Pop, The Generalized Minimum Spanning Tree Problem, PhD thesis, University of Twente, The Netherlands, 2002.
[10] J.J. Prisco, Fiber optic regional area networks in New York and Dallas, IEEE Journal on Selected Areas in Communication SAC-4 (1986) 750–757.
[11] M. Yannakakis, Expressing combinatorial optimization problems by linear programs, Journal of Computer and System Sciences 43 (1991) 441–466.