CARPATHIAN J. MATH.
25 (2009), No. 1, 104 - 118

Online version available at http://carpathian.ubm.ro
Print Edition: ISSN 1584 - 2851 Online Edition: ISSN 1843 - 4401

A survey of different integer programming formulations of the generalized minimum spanning tree problem

P. C. POP

ABSTRACT.
In this survey paper, we discuss the development of the Generalized Minimum Spanning Tree Problem, denoted by GMSTP, and we focus on the integer programming formulations of the problem. The GMSTP is a variant of the classical minimum spanning tree problem (MST), in which the nodes of an undirected graph are partitioned into node sets (clusters) and we are looking for a minimum cost tree spanning a subset of nodes which includes exactly one node from every cluster. In this paper we describe twelve distinct formulations of the GMSTP based on integer programming. Apart from the standard formulations all the new formulations that we describe are 'compact' in the sense that the number of constraints and variables is a polynomial function of the number of nodes in the problem. Comparisons of the polytopes corresponding to their linear relaxations are established.

1. INTRODUCTION

Classical combinatorial optimization problems can be generalized in a natural way by considering a related problem relative to a given partition of the nodes of the graph into node sets. In the literature one finds generalized problems such as the generalized minimum spanning tree problem, the generalized travelling salesman problem, the generalized Steiner tree problem, the generalized (subset) assignment problem, etc. These generalized problems typically belong to the class of NP-complete problems, are harder than the classical ones and nowadays are intensively studied due to the interesting properties and applications in the real world.

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

The GMSTP was introduced by Myung et al. [14] and it has several real world applications in designing metropolitan area networks [7] and regional area networks [24], in determining the location of the regional service centers [17], energy transportation [15], agricultural irrigation [1], etc.

Myung et al. [14] proved that the GMSTP is an NP-hard problem by reduction from the vertex cover problem. A stronger result regarding the complexity of the problem has been provided by Pop et al. [18], namely the GMSTP even defined on trees is NP-hard. The proof of this result was based on a polynomially reduction of the set cover problem to the GMSTP defined on trees. In the same paper [18], the authors presented an exponential time algorithm that finds an exact solution to the GMSTP based on dynamic programming.

A negative approximation result have been proved by Myung *et al.* [14], namely there is no polynomial time algorithm solving the problem with bounded worst-case ratio, unless $P = NP$. Some positive approximation results for the special cases of GMSTP are described in Pop *et al.* [20] providing a polynomial time approximation algorithm with worst-case ratio bounded by 2ρ if the cluster size is bounded by ρ and the cost function is strict positive and satisfies the triangle inequality and in Feremans *et al.* [5] in the case of a geometric version of the GMSTP with grid clustering providing a polynomial time approximation scheme (PTAS) for the case when all non-empty grid cells are connected.

Myung et al. [14] used a branch and bound procedure in order to solve the GMSTP. Their lower procedure is a heuristic method which approximates the linear programming relaxation associated with the dual of the multicommodity flow formulation of the GMSTP. They developed also a heuristic algorithm which finds a primal feasible solution for the GMSTP using the obtained dual solution and reported the exact solution of instances with up to 100 vertices. The GMSTP was solved to optimality for nodes up to 200 by Feremans [2] using a branch-and-cut algorithm. More recently, Pop et al. [21] have proposed a new integer programming formulation of the GMSTP based on a distinction between local and global variables and a solution procedure that finds an optimal solution on instances with up to 240 vertices.

The difficulty of obtaining optimum solutions for the GMSTP has led to the development of several metaheuristics. The first such algorithms were the tabu search (TS) heuristic of Feremans [2] and the simulated annealing (SA) heuristic of Pop [17], an improved version of the SA was described in [23]. Two variants of a TS heuristic and four variable neighborhood search (VNS) based heuristics were later devised by Ghosh [8]. Another VNS algorithm for the GMSTP was proposed by Hu et al. [12]. The authors report that their VNS approach can produce solutions that

Received: 05.11.2008; In revised form: 21.01.2009; Accepted: 30.03.2009
2000 *Mathematics Subject Classification.* 90C27, 90C10, 90C11.
Key words and phrases. *Minimum spanning tree problem, generalized minimum spanning tree problem, integer programming, linear relaxation.*

are comparable to those obtained by means of the second variant of the TS heuristic of Ghosh [8]. The same authors proposed an improved VNS algorithm combining it with integer linear programming [13]. Golden et al. [10] have devised a local search heuristic (LSH) and a genetic algorithm (GA) for the GMSTP. Both algorithms have yielded improvements on TSPLIB instances with sizes between $198 \leq n \leq 225$. On none of these instances did the LSH outperform the GA. Another TS heuristic for the GMSTP was devised by Wang et al. [28]. The authors note that their TS heuristic produces solutions slightly better than those obtained by the GA of Golden et al. [10]. Recently, an attribute based tabu search heuristic employing new neighborhoods was proposed by Oncan et al. [15]. The authors mention that their TS based heuristic yields the best results for all instances. Another metaheuristic approach may be found in [16].

The aim of this paper is to describe twelve different integer programming formulations of the GMSTP and to establish relations between the polytopes corresponding to their linear relaxations.

2. Definition of the generalized minimum spanning tree problem

Let $G = (V, E)$ be an n-node weighted undirected graph whose edges are associated with non-negative costs. 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 $l \neq k$). We denote the cost of an edge $e = (i, j) \in E$ by c_{ij}. Let V_1 be the root cluster, and let $e = (i, j)$ be an edge with $i \in V_l$ and $j \in V_k$. If $l \neq k$, then e is called an *inter-cluster* edge; otherwise, e is called an *intra-cluster* edge.

The *generalized minimum spanning tree problem* (GMSTP) asks for finding a minimum-cost tree T spanning a subset of nodes which includes *exactly* one node from each cluster V_i, $i \in \{1, ..., m\}$. We will call such a tree a *generalized spanning tree*. In the next figure we present examples of generalized spanning trees.

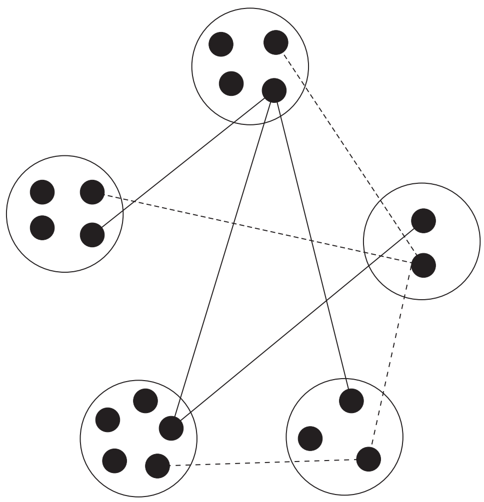

FIGURE 1. Examples of generalized spanning trees

There were considered two variations of the GMSTP in the literature:

- the at least version of the GMSTP in which we are looking for a minimum-cost tree spanning a subset of nodes which includes *at least* one node from each cluster. This variation was introduced by Dror et al. [1], more information on the problem can be found in [2, 15].
- the prize-collecting generalized minimum spanning tree problem (PC-GMSTP) is a variation of the GMSTP in which the nodes within the same cluster are competing to be selected and each node offers a certain prize if selected. More information on this version of the GMSTP can be found in [22, 26, 11].

In the present paper we confine ourselves to the problem of choosing exactly one vertex per cluster.

3. Integer programming formulations of the GMSTP

In order to formulate the GMSTP as an integer program we introduce the following binary variables:

$$x_e = x_{ij} = \begin{cases} 1 & \text{if the edge } e = (i, j) \in E \text{ is included in the selected subgraph} \\ 0 & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} 1 & \text{if the node } i \text{ is included in the selected subgraph} \\ 0 & \text{otherwise} \end{cases}$$

$$w_{ij} = \begin{cases} 1 & \text{if the arc } (i, j) \in A \text{ is included in the selected subgraph} \\ 0 & \text{otherwise.} \end{cases}$$

We use the vector notations $x = (x_{ij})$, $z = (z_i)$, $w = (w_{ij})$ and the notations $x(E^{'}) = \sum_{\{i,j\} \in E'} x_{ij}$, for $E^{'} \subseteq E$, $z(V^{'}) = \sum_{i \in V'} z_i$, for $V^{'} \subseteq V$ and $w(A^{'}) = \sum_{(i,j) \in A'} w_{ij}$, for $A^{'} \subseteq A$.

3.1. **Formulations based on tree properties.**

Generalized subtour elimination formulation (Myung et al. [14])

We consider $G = (V, E)$ an undirected graph. A feasible solution of the GMSTP can be seen as a cycle free graph with $m - 1$ edges, one node selected from each cluster and connecting all the clusters. Therefore the GMSTP can be formulated as the following integer programming problem:

$$\min \sum_{e \in E} c_e x_e$$
$$\text{s.t.} \quad z(V_k) = 1, \quad \forall\, k \in K = \{1, ..., m\} \tag{3.1}$$
$$x(E(S)) \leq z(S - i), \ \forall\, i \in S \subset V, 2 \leq |S| \leq n - 1 \tag{3.2}$$
$$x(E) = m - 1 \tag{3.3}$$
$$x_e \in \{0, 1\}, \quad \forall\, e \in E \tag{3.4}$$
$$z_i \in \{0, 1\}, \quad \forall\, i \in V. \tag{3.5}$$

For simplicity we used the notation $S - i$ instead of $S \setminus \{i\}$. In the above formulation, constraints (3.1) guarantee that from every cluster we select exactly one node, constraints (3.2) eliminate all the subtours and finally constraint (3.3) guarantees that the selected subgraph has $m - 1$ edges. This formulation, introduced by Myung [14], is called the *generalized subtour elimination formulation* since constraints (3.2) eliminate all the cycles.

We denote the feasible set of the linear programming relaxation of this formulation by P_{sub}, where we replace the constraints (3.4) and (3.5) by $0 \leq x_e, z_i \leq 1$, for all $e \in E$ and $i \in V$.

Generalized cutset formulation (Myung et al. [14])

We may replace the subtour elimination constraints (3.2) by connectivity constraints, resulting the *generalized cutset formulation* introduced in [14]:

$$\min \sum_{e \in E} c_e x_e$$
$$\text{s.t.} \quad (1), (3), (4), (5) \text{ and}$$
$$x(\delta(S)) \geq z_i + z_j - 1, \ \forall\, i \in S \subset V, j \notin S \tag{3.6}$$

where the cutset $\delta(S)$ is defined as usually $\delta(S) = \{e = (i, j) \in E \mid i \in S, j \notin S\}$, $S \subset V$.

We denote the feasible set of the linear programming relaxation of this formulation by P_{cut}. Myung et al. [14] proved the following relation: $P_{sub} \subset P_{cut}$.

Generalized multicut formulation (Pop [17])

Our next model, the so-called *generalized multicut formulation*, is obtained by replacing simple cutsets by multicuts. Given a partition of the nodes $V = C_0 \cup C_1 \cup \ldots \cup C_k$, we define the multicut $\delta(C_0, C_1, ..., C_k)$ to be the set of edges connecting different C_i and C_j. The generalized multicut formulation for the GMSTP is:

$$\min \sum_{e \in E} c_e x_e$$
$$\text{s.t.} \quad (1), (3), (4), (5) \text{ and}$$
$$x(\delta(C_0, C_1, ..., C_k)) \geq \sum_{j=0}^{k} z_{i_j} - 1, \ \forall\, C_0, C_1, ..., C_k \text{ node partitions of } V \text{ and } \forall\, i_j \in C_j \text{ for } j = 0, 1, ..., k. \tag{3.7}$$

Let P_{mcut} denote the feasible set of the linear programming relaxation of this model. Clearly, $P_{mcut} \subseteq P_{cut}$, in addition, Pop [17] showed that $P_{sub} = P_{mcut}$.

Cluster subpacking formulation (Feremans [2])

We may strengthen the generalized subtour formulation of the GMSTP by replacing the subtour elimination constraints (3.2) with the *cluster subpacking constraints*

$$x(E(S)) \leq z(S \setminus V_k), \ \ \forall S \subset V, 2 \leq |S| \leq n - 1, k \in K.$$

We observe that these constraints are dominated by:

$$x(E(S')) \leq z(S' \setminus V_k) = z(S') - 1,$$

where $S' = S \cup V_k$. Therefore we arrive at the *cluster subpacking formulation* of the GMSTP, introduced by Feremans [2]:

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

$$s.t. \quad (1), (3), (4), (5) \text{ and}$$

(3.8) $$x(E(S)) \leq z(S) - 1, \ \forall\, S \subset V, 2 \leq |S| \leq n-1, |\{k : V_k \subseteq S\}| \neq 0.$$

The cluster subpacking constraints (3.8) guarantee that the number of edges selected from any subset of nodes S with $S \subset V, 2 \leq |S| \leq n-1$, cannot be greater than the number of nodes selected from that set minus 1. We will denote by P_{spack} the feasible set of the linear programming relaxation of the cluster subpacking formulation.

3.2. Formulations based on arborescence properties.

Consider the directed graph $D = (V, A)$ obtained by replacing each edge $e = (i, j) \in E$ by the opposite arcs (i, j) and (j, i) in A having the same weight as the edge $(i, j) \in E$. The directed version of the GMSTP introduced by Myung et al. [14], called *Generalized Minimum Spanning Arborescence problem* is defined on the directed graph $D = (V, A)$ rooted at a given cluster, say V_1 without loss of generality, and consists of determining a minimum cost arborescence which includes exactly one node from every cluster.

The next two formulations that we are going to present in this section, were presented by Feremans *et al.* [2].

Directed generalized cutset formulation (Feremans [2])

We consider first a *directed generalized cutset formulation* of the GMSTP. In this model we consider the directed graph $D = (V, A)$ with the cluster V_1 chosen as a root, without loss of generality, and we denote $K_1 = K \setminus \{1\}$.

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

$$s.t. \quad z(V_k) = 1, \qquad \forall\, k \in K = \{1, ..., m\}$$

$$x(E) = m - 1$$

(3.9) $$w(\delta^-(S)) \geq z_i, \qquad \forall\, i \in S \subseteq V \setminus V_1$$

(3.10) $$w_{ij} \leq z_i\ , \qquad \forall\, i \in V_1, j \notin V_1$$

(3.11) $$w_{ij} + w_{ji} = x_e\ , \qquad \forall\, e = (i, j) \in E$$

(3.12) $$x, z, w \in \{0, 1\}.$$

In this model constraints (3.9) and (3.10) guarantee the existence of a path from the selected root node to any other selected node which includes only the selected nodes. Let P_{dcut} denote the projection of the feasible set of the linear programming relaxation of this model into the (x, z)-space.

Another possible directed generalized cutset formulation considered by Myung *et al.* in [14], was obtained by replacing (3.3) with the following constraints:

(3.13) $$w(\delta^-(V_1)) \ = \ 0$$

(3.14) $$w(\delta^-(V_k)) \ \leq \ 1, \quad \forall\, k \in K_1.$$

Directed subpacking formulation (Feremans [2])

We introduced now a formulation of the GMSTP based on branchings. Consider, as in the previous formulation, the digraph $D = (V, A)$ with V_1 chosen as the cluster root. The *directed subpacking formulation* of the GMSTP is defined as follows:

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

$$s.t. \quad z(V_k) = 1, \qquad \forall\, k \in K = \{1, ..., m\}$$

$$x(E) = m - 1$$

(3.15) $$w(A(S)) \leq z(S - i), \ \forall\, i \in S \subset V, 2 \leq |S| \leq n - 1$$

(3.16) $$w(\delta^-(j)) = z_j\ , \qquad \forall\, j \in V \setminus V_1$$

$$w_{ij} + w_{ji} = x_e\ , \qquad \forall\, e = (i, j) \in E$$

$$x, z, w \in \{0, 1\}.$$

Let P_{dspack} denote the projection of the feasible set of the linear programming relaxation of this model into the (x, z)-space. Obviously, $P_{dspack} \subseteq P_{sub}$ and $P_{spack} = P_{dspack}$ (see [2]). This means that a simple undirected formulation can be as tight as a directed formulation, even if there are fewer variables. This is also the case for the MST and Steiner Tree Problem.

The following result was established by Feremans [2]: $P_{dspack} = P_{dcut} \cap P_{sub}$. A different proof of this result was presented by Pop [17].

3.3. **Flow based formulations.** All the formulations that we have described so far have an exponential number of constraints. The formulations that we are going to consider next will have only a polynomial number of constraints but an additional number of variables. In order to give compact formulations of the GMST problem one possibility is to introduce 'auxiliary' flow variables beyond the natural binary edge and node variables.

We wish to send a flow between the nodes of the network and view the edge variable x_e as indicating whether the edge $e \in E$ is able to carry any flow or not. We consider four such flow formulations: a single commodity model, a multicommodity model, a bidirectional flow model and a flow cut formulation. In each of these models, although the edges are undirected, the flow variables will be directed. That is, for each edge $(i,j) \in E$, we will have flow in the both directions i to j and j to i.

Single commodity flow formulation (Pop [17])

In the *single commodity model*, the source cluster V_1 sends one unit of flow to every other cluster. Let f_{ij} denote the flow on edge $e = (i,j)$ in the direction i to j. This leads to the following formulation:

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
s.t. \quad & z(V_k) = 1, \qquad \forall\, k \in K = \{1, ..., m\} \\
& x(E) = m - 1 \\
& \sum_{e \in \delta^+(i)} f_e - \sum_{e \in \delta^-(i)} f_e = \begin{cases} (m-1)z_i & \text{for } i \in V_1 \\ -z_i & \text{for } i \in V \setminus V_1 \end{cases} && (3.17) \\
& f_{ij} \le (m-1)x_e, \qquad \forall\, e = (i,j) \in E && (3.18) \\
& f_{ji} \le (m-1)x_e, \qquad \forall\, e = (i,j) \in E && (3.19) \\
& f_{ij}, f_{ji} \ge 0, \qquad \forall\, e = (i,j) \in E && (3.20) \\
& x, z \in \{0,1\}.
\end{aligned}$$

In this model, the mass balance equations (3.17) imply that the network defined by any solution (x,z) must be connected. Since the constraints (3.1) and (3.3) state that the network defined by any solution contains $m-1$ edges and one node from every cluster, every feasible solution must be a generalized spanning tree. Therefore, when projected into the space of the (x,z) variables, this formulation correctly models the GMSTP. We let P_{flow} denote the projection of the feasible set of the linear programming relaxation of this model into the (x,z)-space.

Multicommodity flow formulation (Myung et al. [14])

A stronger relaxation is obtained by considering multicommodity flows. This *directed multicommodity flow model* was introduced by Myung et al. in [14]. In this model every node set $k \in K_1$ defines a commodity. One unit of commodity k originates from V_1 and must be delivered to node set V_k. Letting f_{ij}^k be the flow of commodity k in arc (i,j) we obtain the following formulation:

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
s.t. \quad & z(V_k) = 1, \qquad \forall\, k \in K = \{1, ..., m\} \\
& x(E) = m - 1 \\
& \sum_{a \in \delta^+(i)} f_a^k - \sum_{a \in \delta^-(i)} f_a^k = \begin{cases} z_i & , i \in V_1 \\ -z_i & , i \in V_k \\ 0 & , i \notin V_1 \cup V_k \end{cases} \quad , k \in K_1 && (3.21) \\
& f_{ij}^k \le w_{ij}, \qquad \forall\, a = (i,j) \in A,\ k \in K_1 && (3.22) \\
& w_{ij} + w_{ji} = x_e\ , \qquad \forall\, e = (i,j) \in E \\
& f_a^k \ge 0, \qquad \forall\, a = (i,j) \in A,\ k \in K_1 && (3.23) \\
& x, z \in \{0,1\}.
\end{aligned}$$

In [14], Myung *et al.* presented a branch and bound procedure to solve the GMSTP. The computational efficiency of such a procedure depends greatly upon how quickly it generates good lower and upper bounds. Their lower bounding procedure was based on the directed multicommodity flow model, since its linear programming relaxation not only provides a tight lower bound but also has a nice structure based on which an efficient dual ascent algorithm can be constructed. They developed also a heuristic algorithm which finds a feasible solution for the GMSTP using the obtained dual solution.

We let P_{mcflow} denote the projection of the feasible set of the linear programming relaxation of this model into the (x, z)-space. Pop [17] showed that the following relation exists between the feasible sets of the linear programming relaxation of these flow models $P_{mcflow} \subseteq P_{flow}$.

Bidirectional flow formulation (Pop [17])

We obtain a closely related formulation by eliminating the variables w_{ij}. The resulting formulation consists of constraints (3.1), (3.3), (3.21), (3.23) plus

$$f^h_{ij} + f^k_{ij} \leq x_e \ , \ \forall \, h, k \in K_1 \text{ and } \forall \, e \in E. \tag{3.24}$$

We refer to this model as the *bidirectional flow formulation* of the GMSTP and let P_{bdflow} denote its set of feasible solutions in (x, z)-space. Observe that since we have eliminated the variables w_a in constructing the bidirectional flow formulation, this model is defined on the undirected graph $G = (V, E)$, even though for each commodity k we permit flows f^k_{ij} and f^k_{ji} in both directions on edge $e = (i, j)$.

In the bidirectional flow formulation, constraints (3.24) which we are called the bidirectional flow inequalities, link the flow of different commodities flowing in different directions on the edge (i, j). These constraints model the following fact: in any feasible generalized spanning tree, if we eliminate edge (i, j) and divide the nodes in two sets; any commodity whose associated node lies in the same set as the root node set does not flow on edge (i, j); any two commodities whose associated nodes both lie in the set without the root both flow on edge (i, j) in the same direction. So, whenever two commodities h and k both flow on edge (i, j), they both flow in the same direction and so one of f^h_{ij} and f^k_{ij} equals zero.

Proposition 3.1. (Pop [17]) $P_{mcflow} = P_{bdflow}$.

Proof. If $(w, x, z, f) \in P_{mcflow}$, using (3.11) we have that

$$f^h_{ij} + f^k_{ji} \leq w_{ij} + w_{ji} = x_e \ , \ \ \forall \, e = (i, j) \in E \text{ and } \forall \, h, k \in K_1.$$

On the other hand, assume that $(x, z, f) \in P_{bdflow}$. By (3.24)

$$\max_h f^h_{ij} + \max_k f^k_{ji} \leq x_e \ \ \forall \, e = (i, j) \in E.$$

Hence we can choose w such that $\max_h f^h_{ij} \leq w_{ij}$ and $x_e = w_{ij} + w_{ji}$ for all $e = (i, j) \in E$. For example take

$$w_{ij} = \frac{1}{2}(x_e + \max_h f^h_{ij} - \max_k f^k_{ji}).$$

Clearly, $(w, x, z, f) \in P_{mcflow}$.

We obtain another formulation, which we refer to as the *undirected multicommodity flow model*, by replacing the inequalities (3.22) by the weaker constraints:

$$f^k_{ij} \leq x_e \ , \quad \text{for every } k \in K_1 \text{ and } e \in E.$$

□

Flow cut formulation (Feremans [2])

The flow cut formulation is closely related to the multicommodity flow formulation. The main difference between the two is that the flow balance constraints (3.21) and constraints (3.22) are replaced by a directed cutset constraint using the Gale [6] necessary and sufficient feasibility condition for the existence of a feasible flow.

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
s.t. \quad & z(V_k) = 1, && \forall \, k \in K = \{1, ..., m\} \\
& w_{ij} + w_{ji} = x_e \ , && \forall \, e = (i, j) \in E \\
& w(\delta^-(V_1)) = 0 && (3.25) \\
& w(\delta^-(V_k)) \leq 1, && \forall k \in K_1 \quad (3.26) \\
& w(\delta^-(S)) \geq z(V_k \cap S) - z(V_1 \cap S), && \forall S \subset V, 1 \leq |S| \leq n-1, k \in K_1 \quad (3.27) \\
& x, z \in \{0, 1\}.
\end{aligned}$$

The directed cutset constraint (3.27) together with constraints (3.1), (3.11), (3.25) and (3.26) guarantees tree design and connectivity between all clusters. We refer to this model as the *flow cut formulation* of the GMSTP and let P_{fcut} denote its set of feasible solutions in (x, z)-space.

The relationship between the polyhedron defined by the linear relaxation of the flow cut formulation of the GMSTP and some of the polyhedrons of the linear relaxations of corresponding formulations is (see [2], [17], [26]):

$$P_{spack} = P_{dspack} = P_{fcut} = P_{mcflow} = P_{bdflow}.$$

3.4. **A model based on Steiner tree properties.** Raghavan [25] have fransformed the GMSTP into a Steiner tree problem which can be solved with any of the specialized algorithms for this problem.

Steiner tree formulation (Raghavan [25])

The *Steiner tree formulation*, introduced by Raghavan [25], is a multicommodity flow formulation for the GMSTP, once the GMSTP is transformed into a Steiner tree problem with degree constraints. The transformation of the GMSTP to a Steiner tree problem is performed using the following steps. Let s_r be an artificial root node, and $T = \{t_1, t_2, ..., t_k\}$ be a set of artificial sink nodes that are required to be in the Steiner tree. Artificial node s_r is connected to each node $\imath \in V$ with zero cost arc (s_r, i) and all nodes $i \in V_k$ of a given cluster $k = 1, ..., m$ are connected with zero cost arc (i, t_k) to the corresponding artificial node t_k. In this new graph, a set of commodities H is also defined so that there is exactly one commodity $h \in H$ originating at node s_r and terminating at node t_k for each $k = 1, ..., m$. The Steiner tree formulation of the GMSTP is described as follows:

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

$$\text{(3.28)} \qquad s.t. \sum_{(i,j) \in A: i \notin V_k, j \in V_k} w_{ij} = 1, \qquad \forall\, k \in K_1$$

$$\text{(3.29)} \qquad w_{ij} + w_{ji} \leq 1, \qquad \forall (i,j) \in A$$

$$\text{(3.30)} \qquad \sum_{a \in \delta^+(i)} f_a^h - \sum_{a \in \delta^-(i)} f_a^h = \begin{cases} -1 & \text{, if } i = O(h) \\ 1 & \text{, if } i = D(h)\ \forall i \in V \text{and} h \in H \\ 0 & \text{, otherwise} \end{cases}$$

$$\text{(3.31)} \qquad 0 \leq f_{ij}^h \leq w_{ij}, \qquad \forall\ (i,j) \in A,\ h \in H$$

$$w \in \{0, 1\}.$$

The Steiner tree formulation of the GMSTP is then the standard multicommodity fow model for the Steiner tree problem with additional degree constraints (3.28) that ensure an in-degree of 1 for each of the clusters ($k \in K$). This formulation ensures the design of a minimum-cost directed Steiner tree, containing root node s_r, all the nodes in T, and exactly one node from each cluster.

We denote by P_{stree} the set of feasible solutions in (x, z)-space of this formulation. In [25], Raghavan has shown that the Steiner tree formulation is equivalent to the multicommodity flow formulation, where the node variables had to be used to indicate whether a node is selected to be in the GMSTP.

3.5. **Local-global formulation of the GMSTP.** Our last formulation was introduced by Pop [17] and aims at distinguishing between *global*, i.e. inter-cluster connections, and *local* ones, i.e. connections between nodes from different clusters. We introduce variables y_{ij} $(i, j \in \{1, ..., m\})$ 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 contracted graph with vertex set $\{V_1, ..., V_m\}$ which we assume to be complete).

Following Yannakakis [27] this polytope, denoted by P_{MST}, can be represented by the following polynomial number of constraints:

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

$$\text{(3.32)} \qquad y_{ij} = \lambda_{kij} + \lambda_{kji}, \text{ for } 1 \leq k, i, j \leq m \text{ and } i \neq j$$

$$\text{(3.33)} \qquad \sum_j \lambda_{kij} = 1, \qquad \text{for } 1 \leq k, i, j \leq m \text{ and } i \neq k$$

$$\text{(3.34)} \qquad \lambda_{kkj} = 0, \qquad \text{for } 1 \leq k, j \leq m$$

$$y_{ij}, \lambda_{kij} \geq 0, \qquad \text{for } 1 \leq k, i, j \leq m.$$

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 are $\lambda_{kij} = 1$, if j is the parent of i when we root the tree at k and 0 otherwise.

The constraints (3.32) 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 (3.33) mean that if we root a spanning tree at k then every node other than node k has a parent and finally constraints (3.34) mean that the root k has no parent.

If the vector y describes a spanning tree on the contracted graph, the corresponding best (w.r.t. minimization of the costs) *local solution* $x \in \{0, 1\}^{|E|}$ can be obtained by one of the following two methods described by Pop [17] using dynamic programming or integer linear programming.

If the 0-1 vector y describes a spanning tree on the contracted graph, the corresponding local solution $x \in \{0,1\}^{|E|}$ that minimizes the costs can be obtained by solving the following integer linear programming problem:

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

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 (Pop et al. [21]):

Proposition 3.2. *If y is the 0-1 incidence vector of a spanning tree of the contracted graph then the polyhedron $P_{local}(y)$ is integral.*

In order to prove that the polyhedron $P_{local}(y)$ is integral, Pop et al. [21] showed that every solution of the linear programming relaxation can be written as a convex combination of solutions corresponding to spanning trees: $(x, z) = \sum \lambda_T (x_T, z_T)$. The proof was done by induction on the support of x, denoted by $supp(x)$ and defined as follows $supp(x) := \{x_e \mid x_e \neq 0,\ e \in E\}$.

Local-global formulation

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

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

This new formulation of the GMSTP was obtained by incorporating the constraints characterizing P_{MST}, with $y \in \{0,1\}$, into $P_{local}(y)$. The local-global formulation of the GMSTP uses three types of variables to define the connections between clusters in the graph. The first group of variables, which we have already used in the previous formulations, are indicator edge variables, x_{ij}. These variables define the use of specific edges between nodes in the graph. The other two groups of variables, y_{lr} and λ_{klr}, on the other hand, define connections between clusters in the graph. The y_{lr} variables indicate whether the solutions includes any of the edges directly connecting clusters l and r. The λ_{klr} variables are used to define K directed trees, one for each of the clusters in the graph.

This new model of the GMSTP has several nice properties that can be used for the development of exact solution procedures [17, 21], metaheuristic algorithms [12, 13], heuristic algorithms [18] and as well heuristic and exact procedures for the PC-GMSTP [26, 11].

Note that although the presented formulations are defined for the GMSTP, these formulations are also valid for the PC-GMSTP, we only need to add the prize of selecting a node from a cluster to the objective function.

4. Conclusions

We have presented a survey of twelve integer programming formulations of the GMSTP, 3 introduced by Myung *et al.*, 4 introduced by Feremans *et al.*, 4 introduced by Pop and one introduced by Raghavan. We have shown that the best linear relaxation is provided by the undirected cluster subpacking formulation, directed cluster subpacking formulation, multicommodity flow formulation, bidirectional flow formulation, flow cut formulation and Steiner tree formulation. Among the described integer programming formulations, the local-global formulation is the most compact in terms of number of variables and number of constraints.

The relationships between the described formulations are depicted in the following figure:

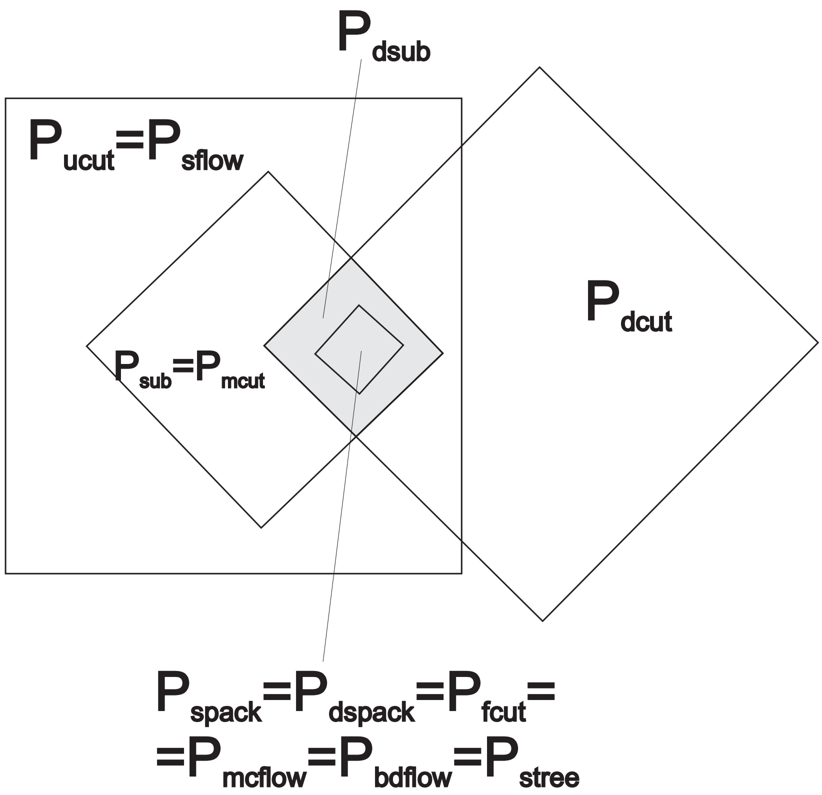

FIGURE 2. Relationship between the polyhedrons defined by the linear relaxations of corresponding GMSTP formulations

REFERENCES

[1] Dror, M., Haouari, M. and Chaouachi, J., *Generalized spanning trees*, European J. Oper. Res. **120** (2000), 583–592

[2] Feremans, C., *Generalized Spanning Trees and Extensions*. PhD thesis, Universite Libre de Bruxelles (2001)

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

[4] Feremans, C., Labbe, M. and Laporte, G., *The generalized minimum spanning tree problem: Polyhedral analysis and branch-and-cut algorithm*, Networks **43** (**2**) (2004), 71–86

[5] Feremans, C., Grigoriev, A. and Sitters, R., *The geometric generalized minimum spanning tree problem with grid clustering*, 4OR, **4** (2006), 319–329

[6] Gale, D., (1957) *A Theorem on Flows in Networks*, Pacific J. Math. **7** (1957), 1073–1082

[7] Gerla, M. and Frata, L., *Tree structured fiber optics MAN's*, IEEE Journal on Selected Areas in Communication **SAC-6** (1988), 934–943

[8] Ghosh, D., *Solving medium to large sized Euclidean generalized minimum spanning tree problems*. Technical Report NEP-CMP-2003-08-02, Indian Institute of Management, Research and Publication Department, Ahmedabad, India (2003)

[9] Ghosh, D., *A Probabilistic Tabu Search Algorithm for the Generalized Minimum Spanning Tree Problem*. Technical Report NEP-CMP-2003-07-02, Indian Institute of Management, Research and Publication Department, Ahmedabad, India (2003)

[10] Golden, B., Raghavan, S. and Stanojevic, D., *Heuristic search for the generalized minimum spanning tree problem*, INFORMS J. Comput. **17** (**3**) (2005), 290–304

[11] Golden, B., Raghavan, S. and Stanojevic, D., *The prize-collecting generalized minimum spanning tree problem*, Journal of Heuristics **14** (2008) , 69–93

[12] Hu, B., Leitner, M. and Raidl, G. R., *Computing Generalized Minimum Spanning Trees with Variable Neighborhood Search*. In Proceedings of the 18th Mini Euro Conference on Variable Neighborhood Search, P. Hansen, N. Mladenovic, J. A. M. Perez, B. M. Batista, and J. M. Moreno-Vega, eds. Tenerife, Spain (2005)

[13] Hu, B., Leitner, M. and Raidl, G. R., *Combining variable neighborhood search with integer linear programming for the generalized minimum spanning tree problem*, J. of Heuristics **14** (5) (2008), 473–499

[14] Myung, Y. S., Lee, C. H. and Tcha, D. W., *On the generalized minimum spanning tree problem*, Networks **26** (1995), 231–241

[15] Oncan, T., Cordeau, J-F. and Laporte, G., *A Tabu Search Heuristic for the Generalized Minimum Spanning Tree Problem*, European J. Oper. Res. **191** (2008), 306–319

[16] Pintea, C. M., Dumitrescu, D. and Pop, P. C., *Combining heuristics and modifying local information to guide and-based search*, Carpathian J. Math. **24** (1) (2008), 94–103

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

[18] Pop, P. C. and Pop Sitar, Corina, *A note on the complexity of the Generalized Minimum Spanning Tree Problem*, Stud. Univ. Babeş-Bolyai Inform., **XLIX** (2004), 2, 75–80

[19] Pop, P. C., Pop Sitar, Corina and Zelina, Ioana, *Efficient Algorithms for the Generalized Minimum Spanning Tree Problem*, Carpathian J. Math. **20** (1) (2004), 109–117

[20] Pop, P. C., Still, G. and Kern, W., *An approximation algorithm for the generalized minimum spanning tree problem with bounded cluster size*. In H. Broersma, M. Johnson, and S. Szeider, editors, Algorithms and Complexity in Durham 2005, Proceedings of the first ACiD Workshop, volume 4 of Texts in Algorithmics, pp 115-121. Kings College Publications (2005)

[21] Pop, P. C., Kern, W. and Still, G., *A New Relaxation Method for the Generalized Minimum Spanning Tree Problem*, European J. Oper. Res. **170** (2006), 900–908
[22] Pop, P. C., *On the prize collecting generalized minimum spanning tree problem*, Ann. Oper. Res. **150** (2007), 193–204
[23] Pop, P. C., Sabo, C., Pop Sitar, Corina and Craciun, M., (2007) *A Simulated Annealing Based Approach for Solving the Generalized Minimum Spanning Tree Problem*, Creat. Math. Inform., **16** (2007), 42–53
[24] Prisco, J. J., *Fiber optic regional area networks in New York and Dallas*, IEEE Journal on Selected Areas in Communication **SAC-4** (1986), 750–757
[25] Raghavan, S., *On modeling the generalized minimum spanning tree*. Technical report, University of Maryland (2002)
[26] Stanojevic, D., *Optimization of contemporary telecommunications networks: generalized spanning trees and WDM optical networks*, PhD thesis, University of Maryland (2005)
[27] Yannakakis, M., *Expressing combinatorial optimization problems by linear programs*, J. Comput. System Sci. **43** (1991), 441–466
[28] Wang, Z., Che, CH. and Lim, A., *Tabu search for generalized minimum spanning tree problem*, Q. Yang and G. Webb (Eds.), PRICAI 2006, LNAI, Springer-Verlag Berlin Heidelberg **4099** (2006), 918–922

DEPARTMENT OF MATHEMATICS
AND COMPUTER SCIENCE
NORTH UNIVERSITY OF BAIA MARE
VICTORIEI 76, 430122 BAIA MARE, ROMANIA
E-mail address: pop_petrica@yahoo.com