A comparison of group Steiner tree formulations.

Young-Soo Myung
Department of Business Administration,
Dankook University, Cheonan, myung@dankook.ac.kr

Abstract

The group Steiner tree problem is a generalization of the Steiner tree problem that is defined as follows. Given a weighted graph with a family of subsets of nodes, called groups, the problem is to find a minimum weighted tree that contains at least one node in each group. We present some existing and some new formulations for the problem and compare the relaxations of such formulations.

1. Introduction

In the group Steiner tree problem (GSTP), we are given an undirected graph *G*=(*V*,*E*) with a nonnegative cost function $c: E \rightarrow R_+$ and subsets of nodes $V_1, V_2, \cdots, V_m$, called groups. Let $K = \{1,2,\cdots,m\}$ be the index set of the groups. There may exist nodes that do not belong to any of the groups. A *group Steiner tree* is defined as a tree that contains at least one node in each group. Then the GSTP is to find a group Steiner tree of minimum cost, where the cost of a tee is the sum of costs of its edges. We assume that the groups are pairwise disjoint but this assumption does not restrict the generality of the problem since we can easily transform a problem not satisfying this assumption to one satisfying it [7]. The GSTP is NP-hard since it is a generalization of the Steiner tree problem (STP). Reich and Widmayer [18] have introduced the GSTP with its applications in VLSI design. Other applications can be found in [2,7,15]. Due to its practical and theoretical significance, a lot of research attention has been given to this problem [1,2,3,4,6,7,11,12,19,21]. If we are restricted to select exactly one node per group, the resulting problem becomes the generalized minimum spanning tree problem (GMSTP). Myung et al. [15] introduced the GMSTP and many researches considered the problem [5,10,17,20]. The GMSTP is also NP-hard. As many of the works on the GSTP and the GMSTP have been done independently, in some papers the name of GMSTP is used to indicate the GSTP and a couple of different names other than the above two were also used. However, both problems can not be trivially transformed to each other.

In this paper, our objective is to describe various integer programming formulations for the GSTP and compare the linear programming (LP) relaxations of them. We consider the formulations already presented in the literature and also introduce some new ones. A comparison of different formulations is an interesting and meaningful subject in combinatorial optimization and similar studies have been done on the STP [9,16] and GMSTP [5].

2. Notation

Throughout the paper, we frequently use the following notation. We refer to undirected graphs as *graphs* and to directed graphs as *digraphs*. In a graph *G*=(*V*,*E*), the elements of *E* are called *edges* and the edge *e* between the node *i* and *j* is denoted by {*i,j*} or {*j,i*}. In a digraph *D*=(*V*,*A*), the elements of *A* are called *arcs* and the arc *a* from node *i* to node *j* is denoted by (*i,j*). (*i,j*) and (*j,i*) do not represent the same arc. Bidirecting an edge *e*={*i,j*} means replacing the edge by two arcs in opposite direction, (*i,j*) and (*j,i*). Given a graph *G*=(*V*,*E*) and a family of subsets of *V*, $S_1, \cdots, S_p$, $\delta_G(S_1, \cdots, S_p)$ represents the set of edges with end nodes in different subsets. For a set of $S \subseteq V$, we also use $\delta_G(S) = \delta_G(S, V-S)$ and use $E_G(S)$ to denote the set of edges in *E* with both end nodes in *S*. When we consider a digraph *D*=(*V*,*A*) we use the following notation. For a set $S \subseteq V, \delta^-_D(S)$ denotes the set of arcs $\{(i,j) \in A : i \not\in S, j \in S\}, \delta^+_D(S) = \delta^-(V-S)$ and $A_D(S) = \{(i,j) : i \in S, j \in S\}$. For exposition brevity, we skip the subscripts *G* and *D*, when the underlying graph or digraph is clear in the context and we write $\delta^-_D(i)$ (resp. $\delta^+_D(i)$ or

$\delta_G(i)$) instead of $\delta_D^-(\{i\})$ (resp. $\delta_D^+(\{i\})$ or $\delta_G(\{i\})$).

If x is defined on the elements of a set M (typically M is an edge set E, an arc set A or a vertex set V) then we denote $\sum_{i \in N} x_i$ for $N \subseteq M$ by $x(N)$. The only exceptions are $\delta(.)$, $\delta^-(.)$, $\delta^+(.)$, $E(.)$ and $A(.)$ which were defined previously.

3. Existing formulations

Because of the similarities among the STP, the GMSTP and the GSTP, we may expect to formulate the GSTP by directly using the formulations for the STP and the GMSTP. Actually, some formulations of the GSTP can be obtained by slightly modifying the ones for the other problems. However, in order to use a certain formulation for the STP and the GMSTP, we have to either know in priori at least one node in a selected tree or to assume that exactly one node is selected per group. For this reason, some researchers transformed the problem into a degree constrained STP [2,11]. In this section, we introduce several formulations for the GSTP that have already appeared in the literature. To describe a selected graph, we define an incidence vector x such that $x_e = 1$ if edge e is included in the selected subgraph and 0 otherwise.

3.1 Multicut based formulation

Given a graph $G=(V,E)$ and a partition $S_1,\cdots,S_p$ of V, that is defined as a set of disjoint subsets of V whose union is V, we call $\delta(S_1,\cdots,S_p)$ a *multicut*. Multicut is a generalization of a simple cut that is defined as $\delta(S)$ for a nonempty set $S \subseteq V$. We will say that a partition $S_1,\cdots,S_p$ of V is a *group-partition*, if for every component S_i of the partition, $V - S_i$ contains at least one group V_k for some $k \in K$, i.e., $S_i \cap V_k = \varnothing$. Ferreira and de Oliveria Filho [6] proposed the following formulation for the GSTP using a class of multicut constraints defined on group-partitions.

$$(mcut)\ \min \sum_{e \in E} c_e x_e \qquad (1)$$

$$s.t.\ x(\delta(S_1,\cdots,S_p)) \geq 1,\ \text{for all group partition } S_1,\cdots,S_p \text{ of } V \qquad (2)$$

$$x_e \geq 0, \qquad \forall e \in E \qquad (3)$$

$$x : \text{integer} \qquad (4)$$

(mcut) is based on the fact that a minimal subgraph having a path between any pair of groups becomes a group Steiner tree. Ferreira and de Oliveria Filho [6] proved that the separation problem for the constraints (2) is NP-complete. Therefore, we can not expect to solve the LP relaxation of (mcut) in polynomial time and furthermore, we will show later that the LP relaxation of (mcut) is not tight compared with other formulations presented in this paper. However, (mcut) is a unique formulation found in the literature that uses only edge variables, that is, it is a natural formulation for the GSTP.

3.2 Node variable based formulations

Although there are no node weights, we can use node variables to describe which nodes are included in the selected subgraph. For this purpose, we define an incidence vector y such that $y_i = 1$ if node i is in the selected subgraph and 0 otherwise. Feremans et al. [4] proposed the following formulation for the GSTP.

$$(\text{sub1})\ \min \sum_{e \in E} c_e x_e$$

$s.t.$ (3),(4) and

$$y(V_k) \geq 1, \qquad \forall k \in K \qquad (5)$$

$$x(E) = y(V) - 1, \qquad (6)$$

$$x(E(S)) \leq y(S) - y_i, \quad \forall i \in S \subseteq V \qquad (7)$$

$$x_e \leq 1, \qquad \forall e \in E \qquad (8)$$

$$y_i \leq 1, \qquad \forall i \in V \qquad (9)$$

$$y_i \geq 0, \qquad \forall i \in V \qquad (10)$$

$$y : \text{integer}$$

The constraints (5) ensure that at least one node is selected per group and the equation (6) implies that the number of edges in a tree equals one less than the number of spanned nodes. The constraints (7) are well known *generalized subtour elimination constraints* that prevent the solution from containing cycles.

Formulations similar to (sub1) are also considered for the STP by Goemans [8] and for the GMSTP by Myung et al. [15]. Salazar [19] have shown that if $V_k \subseteq S$ for some $k \in K$, generalized subtour elimination constraints (7) for S can be replaced by $x(E(S)) \leq y(S) - 1$. Using this observation, Salazar proposed the following formulation.

$$(\text{sub2}) \ \min \sum_{e \in E} c_e x_e$$

$$s.t. \quad (3),(4),(5),(6),(8),(9),(10),(11) \text{ and}$$

$$x(E(S)) \le y(S) - y_i, \quad i \in S \in \Sigma_1 \quad (12)$$

$$x(E(S)) \le y(S) - 1, \quad S \in \Sigma_2 \quad (13)$$

where $\Sigma_1 = \{S \subseteq V | S \not\supseteq V_k \forall k \in K\}$ and $\Sigma_2 = \{S \subseteq V | S \supseteq V_k, \text{for some } k \in K\}$.

Both formulations (sub1) and (sub2) are equivalent but the LP relaxations of them in which the integer constraints (4) and (11) are omitted, provide different objective values. We will analyze them in the next section.

3.3 Degree constrained STP formulations

The GSTP can be transformed to a degree constrained STP. Suppose that G=(V,E) with a set of groups $V_1, \cdots, V_m$ and a nonnegative cost function c are given as an instance of the GSTP. We add one dummy node per group and edges between a dummy node and each node in the group with which the dummy node is associated. We assume 0 edge weight for all newly added edges. Let $T = \{n+1, \cdots, n+m\}$ be the set of dummy nodes where node $n+k$ is assigned to group V_k and let E_k for $k \in K$ be the set of edges between node $n+k$ and the nodes in V_k. Let $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ be the augmented graph where $\widetilde{V} = V \cup T$ and $\widetilde{E} = E \cup E_1 \cup \cdots \cup E_m$. Consider the following degree constrained STP, in which our objective is to select a minimum cost tree connecting dummy nodes such that edges adjacent to each dummy node is at most one. In this problem, all nodes in V are Steiner nodes that may or may not be contained in the selected tree. It is not difficult to know that there exists a one-to-one match between a degree constrained Steiner tree of $\widetilde{G}$ and a group Steiner tree of G and both trees have the same objective value.

Duin et al. [3] considered the degree constrained STP to develop a heuristic of solving the GSTP. We can obtain various formulations for the degree constrained STP from those for the STP by simply adding the constraints forcing each dummy node to have a single degree. In the literature, a myriad of formulations for the STP have been proposed and a comparison of their LP relaxations was also well studied in [9,16]. Houari and haouachi [11] presented three formulations for the GSTP in such a way. Here, we introduce a flow based formulation appeared in [11]. To construct a tight formulation, they describe it on a digraph that is obtained by bidirecting each edge of a given graph. Let A be the set of arcs obtained by bidirecting the edges of E, i.e., $A = \{(i,j) | \{i,j\} \in E\}$. For the edges in E_k, $k \in K$, we replace each edge by one arc because we don't need the incoming arcs to the root dummy node and the outgoing arcs from the remaining dummy nodes due to the degree constaints. We set node n+1$ as the root node and define $A_1 = \{(n+1, j) | j \in V_1\}$ and $A_k = \{(j, n+k) | j \in V_k\}$ for $k \in K - \{1\}$. Let $D = (\widetilde{V}, \widetilde{A})$ denote the resulting digraph where $\widetilde{A} = A \cup A_1 \cup \cdots \cup A_m$ and let $K_1 = K - \{1\}$. In a directed graph, each arc replacing the edge e has the cost c_e.

Our objective is to select a minimal digraph having a directed path from node n+1 to node n+k, for each $k \in K_1$. We additionally define an incidence vector w such that $w_a = 1$ if arc a is included in the subgraph and 0 otherwise. The flow formulation considers w_a as the capacity of arc a and determines the variables such that one unit of flow can be sent from node n+1 to node n+k for each $k \in K_1$, which we will call the flow of commodity k. So, we additionally need flow variables f_a^k representing the flow of commodity k in arc a. Then the following flow based formulation describes the GSTP.

$$(\text{dcs}) \ \min \sum_{e \in E} c_e x_e$$

$$s.t. \quad w(\delta^+(n+1)) = 1, \quad (14)$$

$$w(\delta^-(n+k)) = 1, \quad (15)$$

$$f^k(\delta^+(j)) - f^k(\delta^-(j)) = \begin{cases} 1, & \text{if } j = n+1 \\ -1, & \text{if } j = n+k \\ 0, & \text{if } j \notin \{n+1, n+k\} \end{cases}, k \in K_1 \quad (16)$$

$$f_a^k \le w_a, \quad a \in \widetilde{A}, k \in K_1 \quad (17)$$

$$x_e = w_{ij} + w_{ji}, \quad e = \{i,j\} \in E \quad (18)$$

$$f_a^k \ge 0, \quad a \in \widetilde{A}, k \in K_1 \quad (19)$$

$$w_a \ge 0, \quad a \in \widetilde{A} \quad (20)$$

$$w : \text{integer} \quad (21)$$

The constraints (16) ensure the existence of a unit flow from the root dummy node to the other dummy nodes. Although we don't need x variables, we insert it for later use when we compare the LP relaxations of the formulations.

4. New compact formulations

In this section, we present three new

formulations, one using node variables and the other two using a dummy node and a degree constraint. The first formulation comes from the observation that the constraints (8), (9) and (12) can be omitted in (sub2). In other words, the following formulation is also valid for the GSTP.

(sub3) $\min \sum_{e \in E} c_e x_e$

$s.t.$ (3),(4),(5),(6),(10),(11),(13)

It is not trivial to show that (sub3) is a valid formulation and we will show it in the next section.

Our next two formulations are motivated by the observation that we can use only one dummy node to describe various formulations that can be obtained via the degree constrained STP. As defined in Section 3.3, node n+1 denotes a dummy node associated with the group V_1 and E_1 denotes the set of edges with 0 weight between the dummy node and each node in V_1. Our two formulations are also based on a digraph. Let A and A_1 be the set of arcs as defined in the previous section. Then our formulations are defined on a digraph, $D=(\widetilde{V},\widetilde{A})$ where $\widetilde{V}=V\cup\{n+1\}$ and $\widetilde{A}=A\cup A_1$. Notice that D is a subgraph of the digraph we considered in Section 3.3. If we restrict the degree of node n+1 to 1, a subgraph having a path from node n+1 to at least one node in each group corresponds to a group Steiner tree. We describe two formulations, one using flow variables and the other using the cut constraint. We use an incidence vector w to identify which arcs to be included in a selected subgraph and for a flow based formulation, we use flow variables f_a^k representing the flow destined for group $k\in K_1$ in arc a.

The following flow based formulation describes the GSTP.

(flow) $\min \sum_{e \in E} c_e x_e$

$s.t.$ (14),(17),(18),(19),(20),(21), and

$$f^k(\delta^+(j)) - f^k(\delta^-(j)) = \begin{Bmatrix} 1, & \text{if } j=n+1 \\ 0, & \text{if } j \notin \{n+1\}\cup V-k \end{Bmatrix}, k\in K_1 \quad (22)$$

The constraints (22) ensure the existence of a unit flow from the root node to each group, which implies that there exist a path from node n+1 to each group in a selected graph.

Our last formulation is the following cut based formulation.

(cut) $\min \sum_{e \in E} c_e x_e$

$s.t.$ (14),(18),(20),(21), and

$$w(\delta^-(S)) \geq 1, \quad S\in \Sigma_2 \quad (23)$$

It is well known that the constraints (22) and (17) can be replaced by the so-called cut constraints (23) by the max-flow min-cut theorem.

Formulation (sub3) has less constraints than both (sub1) and (sub2) and formulations (flow) and (cut) than similar formulations based on the degree constrained STP. So our new formulations are more compact than the corresponding ones presented in Section 3. Nevertheless, the LP relaxations of our new formulations provide lower bounds as good as any other existing formulations. We will show it in the next section. Ferreira and de Oliveria Filho [6] also proposed a formulation using flow variables and another one using the cut constraints but both of them were proved to be invalid by Myung [13].

5. A comparison of LP relaxations

In this section, we compare the LP relaxations of the formulations considered in Section 3 and 4. The optimal objective value of the LP relaxation of each formulation becomes a lower bound for the GSTP. If the computing time for solving the LP relaxations of the formulations we consider are same, the one giving higher lower bound is most preferable. We will compare the formulations in terms of the lower bounds they can provide. If $(\cdot)$ is an integer programming formulation presented in Section 3 and 4, we let $F(\cdot)$ denote a feasible region of its LP relaxation where the integrality restriction on the variable, (4),(11) and (21) are removed and $F_x(\cdot)$ is the projection of $F(\cdot)$ in the space of x variables. We also use $v(\cdot)$ as the optimal objective value of the LP relaxation.

Among the formulations presented in this paper, the following relations hold.

Theorem 1 For an arbitrary instance of the GSTP with a nonnegative cost function c, v(sub2)=v(sub3)=v(dcs)=v(cut)=v(flow), v(sub1)$\leq$ v(sub2), and v(mcut)$\leq$ v(sub2). And there exist an instance for which v(mcut)<v(sub1) and one for which v(mcut)>v(sub1).

The proof of the theorem is very lengthy and omitted here. The theorem also implies the following fact.

Corollary 2 (sub3) is a valid formulation for the GSTP.

References

[1] C. Chekuri, G. Even, G. Kortsarz, A geedy approximation algorithm for the group Steiner tree problem, Discrete Applied Mathematics 154 (2006) 15-34.

[2] M. Dror, M. Houari, J. Chaouachi, Generalized spanning trees, European Journal of Operational Research 120 (2000), 583-592.

[3] C.W. Duin, A. Volgenant, S. Voβ, Solving group Steiner tree problems as Steiner problems, European Journal of Operational Research 154 (2004) 323-329.

[4] C. Feremans, M. Labbe, and G. Laporte, On generalized minimum spanning tree problem, European Journal of Operational Research 134 (2001), 457-458.

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

[6] C.E. Ferreira, F.M. de Oliveria Filho Chen, Some formulations for the group Steiner tree problem, Discrete Applied Mathematics 154 (2006) 1877-1884.

[7] N. Garg, G. Konjevod, R. Ravi, A polylogarithmic approximation algorithm for the group Steiner tree problem, Journal of Algorithm 37 (2000) 66-84.

[8] M.X. Goemans, The Steiner tree polytope and related polyhedra, Mathematical Programming 63 (1994), 157-182.

[9] M.X. Goemans and Y.-S. Myung, A catalog of Steiner tree formulations, Networks 23 (1993), 19-28.

[10] B. Golden, S. Raghavan, and D. Stanojevic, Heuristic search for the generalized minimum spanning tree problem, INFORMS Journal on Computing 17 (2005), 290-304.

[11] M. Houari and J.S. Chaouachi, Upper and lower bounding strategies for the generalized spanning tree problem, European Journal of Operational Research 171 (2006), 632-647.

[12] E. Ihler, G. Reich, P. Widmayer, Class Steiner trees and VLSI-design, Discrete Applied Mathematics 154 (1999) 173-194.

[13] Y.-S. Myung, A note on some formulations for the group Steiner tree problem, Working paper, Dankook Univ. (2007).

[14] Y.-S. Myung, A group Steiner tree formulation with generalized subtour elimination constraints (in Korean), Working paper, Dankook Univ. (2008).

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

[16] T. Polzin and S.V. Daneshmand, A comparison of Steiner tree relaxaions, Discrete Applied Mathematics 112 (2001), 241-261.

[17] P.C. Pop, W. Kern, and G. Still, A new relaxation method for the generalized minimum spanning tree problem, European Journal of Operational Research 170 (2006), 900-908.

[18] G. Reich, P. Widmayer, Beyond Steiner's problems: a VLSI oriented generalization, Proceedings of Graph-Theoretic Concepts in Computer Science (WG-89), Lecture Notes in Computer Science, Vol 411, Springer, Berlin, 1990, 196-210.

[19] J.J. Salazar, A note on the generalized Steiner tree polytope, Discrete Applied Mathematics 100 (2000) 137-144.

[20] Z. Wang, C.H. Che, and A. Lim, Tabu Search for Generalized Minimum Spanning Tree Problem, Lecture Notes in Computer Science 4099, Springer, Berlin, 2006.

[21] B. Yang, P. Gillard, The class Steiner minimal tree problem: a lower bound and test rpoblem generation, Acta Informatica 37 (2000) 193-211.