HEURISTIC ALGORITHMS FOR THE GENERALIZED MINIMUM SPANNING TREE PROBLEM

by
Petrică C. Pop
Ioana Zelina

Abstract. We consider a generalization of the classical minimum spanning tree problem called the generalized minimum spanning tree problem and denoted by GMST problem. The aim of this paper is to present an exact exponential time algorithm for the GMST problem as well three heuristic algorithms, two of them based on Prim's and Kruskal's algorithms for the minimum spanning tree problem and one based on the local global approach. These three algorithms are implemented and computational results are reported for many instances of the problem.
Keywords: graph theory, heuristic algorithms, complexity theory, generalized minimum spanning trees, Prim's algorithm, Kruskal's algorithm, dynamic programming.

1. Introduction

Many combinatorial optimization problems are *NP-hard,* and the theory of *NP*-completeness has reduced hopes that *NP-hard* problems can be solved within polynomially bounded computation times. Nevertheless, sub-optimal solutions are sometimes easy to find. Consequently, there is much interest in approximation and heuristic algorithms that can find near optimal solutions within reasonable running time.

In mathematical programming, a *heuristic method* or *heuristic* for short is a procedure that determines good or near-optimal solutions to an optimization problem. As opposed to exact methods, heuristics carry no guarantee that an optimal solution will be found.

Practically, for many realistic optimization problems good solutions can be found efficiently, and heuristics are typically among the best strategies in terms of efficiency and solution quality for problems of realistic size and complexity. Heuristics can be classified as either *constructive* (greedy) or as *local search* heuristics. The former are typically one-pass algorithms whereas the latter are strategies of *iterative improvement.*

Useful references on heuristic methods can be found in Osman and Laporte [7] and Reeves [12] 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.

The model fits various problems of determining the location of regional service centers (e.g. public facilities, branches, distribution centers) which should be connected by building links (e.g. highways, communication links). For example, when a company tries to establish marketing centers, one for each market segment, and construct a communication network which interconnects the established centers, the company faces a GMST problem. For another example, when designing metropolitan area networks [4] and regional area networks [11], we are to interconnect a number of local area networks. For this model, we must select a node in each local network as a hub (or a gateway) and connect the hub nodes via transmission links such as optical fibers. Then, such a network design problem reduces to a GMST problem.

The aim of this paper is to present an exact exponential time algorithm for the GMST problem based on dynamic programming as well as to present three constructive heuristic algorithms, two of them based on Prim's and Kruskal's algorithms for the minimum spanning tree problem and one based on the local global approach.

The GMST problem has been introduced by Myung, Lee and Tcha in [6]. Feremans, Labbe and Laporte [3] and P.C. Pop [9] present several integer formulations of the GMST problem and compare them in terms of their linear programming relaxations.

In [9], P.C. Pop presented an in-approximability result for the GMST problem: under the assumption $P \neq NP$, there is no approximation algorithm for the GMST problem. However, under the following assumptions: the graph has bounded cluster size, and the cost function is strictly positive and satisfies the triangle inequality, i.e. $c_{ij} + c_{jk} \geq c_{ik}$ for all $i, j, k \in V$ a polynomial approximation algorithm for the GMST problem was presented.

2. Definition and Complexity of the Problem

The GMST problem is defined on an undirected graph $G = (V, E)$ with nodes partitioned into m clusters. Let $|V| = n$ and $K = \{1, 2,..., m\}$ be the index set of the node sets (clusters). Then, we have
$V = V_1 \cup V_2 \cup ... \cup V_m$ and $V_l \cap V_k = \emptyset$ for all $l, k \in K$ such that $l \neq k$. We assume that edges are defined only between nodes belonging to different clusters and each edge $e = \{i, j\} \in E$ has a nonnegative cost denoted by c_{ij} or by $c(i, j)$.

The GMST problem is the problem of finding 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.

Myung *et al.* in [6] proved that the GMST problem is *NP-hard.* We prove now a stronger result regarding the complexity of the problem.

Garey and Johnson [3] have shown that for certain combinatorial optimization problems, the simple structure of trees can offer algorithmic advantages for efficiently solving them. Indeed, a number of problems that are *NP-complete,* when formulated on a general graph, become polynomially solvable when the graph is a tree. Unfortunately, this is not the case for the GMST problem. We will show that on trees the GMST problem is *NP-hard.*

Let us consider the case when the GMST problem is defined on trees, i.e. the graph $G = (V, E)$ is a tree. To show that the GMST problem on trees is *NP-hard* we introduce the so-called *set cover problem* which is known to be *NP-complete* (see [3]).

Given a finite set $X = \{ x_1 ,..., x_a \}$, a collection of subsets, $S_1 ,... ,S_b \subseteq X$ and an integer $k < |X|$, the *set cover problem* is to determine whether there exists a subset $Y \subseteq X$ such that $|Y| \leq k$ and

$$S_c \cap Y \neq \emptyset, \ \forall c \text{ with } 1 \leq c \leq b.$$

We call such a set Y a *set cover* for X.

The following result holds:

Theorem 1. *The Generalized Minimum Spanning Tree problem on trees is NP-hard.*

The proof of this theorem is based on the fact that the set cover problem can be polynomially reduced to the GMST problem defined on trees. The complete proof may be found in [9].■

3. An Exact Algorithm for the GMST Problem

In this section, we present an algorithm that finds an exact solution to the GMST problem based on dynamic programming.

Let G' be the graph obtained from G after replacing all nodes of a cluster V_i with a supernode representing V_i, that we will call the *global graph.* For convenience, we identify V_i with the supernode representing it. We assume that G' with vertex set $\{V_1, ..., V_m\}$ is complete. Given a global spanning tree of G', which we shall refer to as the *global spanning tree,* we use dynamic programming in order to find the best (w.r.t. cost minimization) generalized spanning tree.

Fix an arbitrary cluster V_{root} as the root of the global spanning tree and orient all the edges away from vertices of V_{root} according to the global spanning tree. A directed edge $\langle V_k, V_l \rangle$ resulting from the orientation of edges of the global spanning tree defines naturally an orientation $\langle i,j \rangle$ of an edge $(i, j) \in E$ where $i \in V_k$ and $j \in V_l$. Let v be a vertex of cluster V_k for some $1 \le k \le m$. All such nodes v are potential candidates to be incident to an edge of the global spanning tree.

Let $T(v)$ denote the subtree rooted at such a vertex v; $T(v)$ includes all vertices reachable from v under the above orientation of the edges of G based on the orientation of the edges of the global spanning tree. The *children* of $v \in V_k$, denoted by $C(v)$, are those vertices $u \in V_l$ which are heads of the directed edges $\langle v, u \rangle$ in the orientation. The leaves of the tree are those vertices that have no children.

Let $W(T(v))$ denote the minimum weight of a generalized subtree rooted at v. We want to compute

$$\min_{r \in V_{root}} W(T(r))$$

We are now ready for giving the dynamic programming recursion to solve the subproblem $W(T(v))$. The initialization is:

$W(T(v)) = 0$, if $v \in V_k$ and V_k is a leaf of the global spanning tree.

To compute $W(T(v))$ for an interior vertex $v \in V$ to a cluster, i.e., to find the optimal solution of the subproblem $W(T(v))$, we have to look at all vertices from the clusters V_l such that $C(v) \cap V_l \neq \emptyset$. If u denotes a child of the interior vertex v, then the recursion for v is as follows:

$$W(T(v)) = \sum_{l, C(v) \cap V_l \neq \phi} \min_{u \in V_l} [c(u,v) + W(T(u))].$$

Hence, for fixed v we have to check at most n vertices. Consequently, for the given global spanning tree, the overall complexity of this dynamic programming algorithm is $O(n^2)$. Since by Cayley's formula [1], the number of all distinct global spanning trees is m^{m-2}, we have established the following:

Theorem 2. *There exists a dynamic progmmming algorithm which provides an exact solution to the GMST problem in* $O(m^{m-2}n^2)$ *time, where n is the number of nodes and m is the number of clusters in the input graph.*

Clearly, the above is an exponential time algorithm unless the number of clusters m is fixed.

4. Heuristic Algorithms for the GMST problem

We present three constructive heuristic algorithms two of them based on Prim's and Kruskal's algorithms for the minimum spanning tree problem and one algorithm based on the local global approach.

The first heuristic algorithm for the GMST problem is based on Kruskal's algorithm and is a greedy algorithm. Given a set of objects, the greedy algorithm attempts to find a feasible subset with minimum objective value by repeatedly choosing an object of minimum cost among the unchosen ones and adding it to the current subset provided that the resulting subset is feasible.

In particular, the heuristic algorithm for the GMST problem based on Kruskal's algorithm, works by repeatedly choosing an edge of minimum cost among the edges not chosen so far and adding this edge to the "current generalized spanning forest" i.e. a forest that spanns a subset of nodes which includes at most one node from each cluster. Whenever an edge is selected all the nodes from the clusters that contain the end nodes of that edge are deleted. The algorithm terminates when the current generalized spanning forest becomes connected.

A Heuristic Algorithm for the GMST problem based on Kruskal's Algorithm

Input: A connected graph $G = (V, E)$ with the nodes partitioned into m clusters $V_1, V_2, ..., V_m$ and edges defined between nodes from different clusters with positive cost.

Output: Edge set $F \subset E$ of minimum generalized spanning tree of G that approximates the optimal solution of the GMST problem.

F : = Φ*; (F is the edge set of the current generalized spanning forest)*
linearly order the edges in E according to nondecreasing cost;
let the ordering be: $e_1, e_2, ..., e_{|E|}$ *;*
for each edge e_i *, i = 1, 2, ..., |E|, do*
    if $F \cup \{e_i\}$ *has no cycle*
        then $F := F \cup \{e_i\}$ *and whenever we choose a node from a cluster we delete all the*
            other nodes from that cluster and all the edges adjacent
        to the deleted nodes;
    if |F| = |V| - 1 then stop and output F;
    end; {if}
    end; {if}
end; {for}

The following obvious result holds:

Theorem 3. *The previous algorithm yields a generalized spanning tree of G that approximates the optimal solution of the GMST problem and its running time is O(mn), where m is the number of clusters and n is the number of nodes of G*.

The second algorithm that we are going to present is based on Prim's algorithm for spanning trees.

This algorithm starts with a "current generalized tree" *T* that consists of a single node. In each iteration, a minimum cost edge in the boundary $\delta(V(T))$ of *T* is added to *T*. Whenever a node is selected all the other nodes from that cluster are deleted. The algorithm ends when *T* is a generalized spanning tree.

A Heuristic Algorithm for the GMST problem based on Prim's Algorithm

Input: A connected graph $G = (V, E)$ with the nodes partitioned into *m* clusters $V_1, V_2, ..., V_m$ and edges defined between nodes from different clusters with positive cost.
Output: A generalized spanning tree $T = (S, F)$ of *G* that approximates the optimal solution of the GMST problem.

F:=$\emptyset$*;*
S:= $\{v_{i_0}\}$, *where* $v_{i_0} \in V_r$, $r \in K$;
while $|S| < m$ *do*
Among all the global edges having exactly one end in S, find an edge $(v_{i_0}, v_j) \in E$ *that minimize*
the cost and whenever we select a node in a cluster delete all the other nodes from that cluster
and all the edges adjacent to the deleted nodes;
$F := F \cup \{(v_{i_0}, v_j)\}$;
$S := S \cup (\{v_{i_0}, v_j\} \setminus S)$;
end; { while}

The following obvious result holds:

Theorem 4. *The previous algorithm yields a generalized spanning tree of G that approximates the optimal solution of the GMST problem and its running time is* $O(n^2)$, *where m is the number of clusters and n is the number of nodes of G.*

A Heuristic Algorithm for the GMST problem based on the local-global approach

The local-global approach aims at distinguishing between *global,* i.e. intercluster connections, and *local* ones*,* i.e. connections between nodes from different clusters.

We call *global graph* denoted by $G^{global} = (V^{global}, E^{global})$ the graph obtained by shrinking all the nodes from every cluster into one, i.e. $V^{global} = \{V_1, ..., V_m\}$, $|V^{global}| = m$. The edges are defined as connections of the clusters and we assume that the global graph is complete.

It is worth to mention that having a tree of the global graph the corresponding best generalized spanning tree w.r.t. minimization of the cost, can be found using the dynamic programming presented in Section 2 or by using integer programming, see [9].

Before presenting the heuristic algorithm we make some notations: let
F^{global} denote the current edge set in the complete global graph;
S^{global} denote the current node set in the complete global graph;

F denote the current edge set in the graph *G;*

S denote the current node set in the graph *G* and

T denote the minimum generalized tree, given the tree global connection of the clusters i.e., F^{global}.

It is easy to observe that the following hold: $|F| = |F^{global}|$ and $|S| = |S^{global}|$.

A Heuristic Algorithm for the GMST problem based on the local-global approach

Input: A connected graph $G = (V, E)$ with the nodes partitioned into m clusters $V_1, V_2, ..., V_m$ and edges defined between nodes from different clusters with positive cost.

Output: A generalized spanning tree $T = (S, F)$ of G that approximates the optimal solution of the GMST problem.

$F^{global} := \emptyset$; $S^{global} := \{V_r\}$, where V_r is an arbitrary cluster;

$F := \emptyset$; $S := \{v_{i_0}\}$, where $v_{i_0} \in V_r$;

while $|S| < m$ do

Among all the global edges having exactly one end in S^{global} find a global edge $(V_k, V_l) \in E^{global}$

that minimize the cost of the corresponding generalized tree;

$F^{global} = F^{global} \cup \{(V_k, V_l)\}$;

$S^{global} := S^{global} \cup (\{V_k, V_l\} \setminus S^{global})$;

$F := \{(v_i, v_j) \mid (v_i, v_j) \in T\}$;

$S := \{ v_i \mid v_i \in V_k \cap T, V_k \in S^{global}\}$;

end;{while}

Theorem 5 *The previous algorithm yields* a *generalized spanning tree of G that approximates the optimal solution of the GMST problem and its running time* is *$O(m^2n^2)$, where m* is *the number of clusters and n* is *the number of nodes of G.*

Proof. The step in the loop can be done in polynomial time by using the dynamic programming algorithm presented in the Section 3. That dynamic programming gives us the new node set S and the edge set F and its running time is $O(n^2)$.

Applying Prim's algorithm for the global graph, the running time for finding a global spanning tree is $O(m^2)$.

Therefore the running time of the heuristic based on the local-global approach is $O(m^2n^2)$.■

5. Computational results

We considered an undirected graph $G=(V,E)$ having n nodes which are partitioned into m clusters such that each cluster has the same number of nodes. Edges are defined between all the nodes from different clusters and the cost of the edges were generated randomly in the interval [0,100]. For each type of instance we considered five trials.

Our algorithms have been implemented in Delphi 5. The computational experiments were performed on a personal computer with AMD Athlon 2400, 1.67 GHz, 512 RAM and SO Windows XP Professional SP1.

In the next table we compare the computational results for solving the problem using the heuristic algorithms with the computational results obtained using the rooting procedure given by Pop [8].

Problem size		Based on Kruskal		Based on Prim		Local-global		Rooting procedure	
m	n	OPT/UB	CPU (msec)	OPT/UB	CPU (msec)	OPT/UB	CPU (sec)	LB/ OPT	CPU (sec)
10	30	73.966	2	80.003	0	98.432	0.1	100	0.1
	40	71.481	4	70.274	0	92.567	0.9	100	0.7
	60	72.250	21	71.339	1	91.198	1.3	100	0.9
12	36	86.498	3	85.490	0	94.974	0.2	100	0.1
	48	82.959	14	82.547	1	90.584	1.9	100	1.6
	72	72.211	67	71.239	3	89.480	7.2	100	5.6

15	45	88.195	7	82.436	1	95.478	0.4	100	0.2
	60	76.707	28	76.385	3	92.834	3.5	100	2.8
	90	72.998	95	84.908	12	90.456	7.1	100	5.9

The first two columns in the table give the size of the problem: the number of clusters *m* and the number of nodes *n*. The next columns describe the Kruskal based heuristic algorithm, the Prim based heuristic based algorithm and the heuristic algorithm based on the local-global approach and contain the upper bounds obtained as a percentage of the optimal value of the GMST problem (OPT/UB) and the computational times (CPU) in miliseconds for solving the GMST problem. The last columns describe the rooting procedure and contain the lower bounds 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, see [8].

As we can see from the table the heuristic algorithms provides us good sub-optimal solution in reasonable time.

References:

[1] Cayley, A.(1874). *On the mathematical theory of isomers*, Philosophical Magazine, 67, 444

[2] Feremans, C., Labbe, M. and Laporte, G. (2000). *A Comparative Analysis of SeveraI Formulations for the Generalized Minimum Spanning Tree Problem*, Technical Report IS-MG 2000/22, Universite Libre de Bruxelles, Belgium, to appear in Networks

[3] Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractibility: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, California

[4] Gerla, M. and Frata, L. (1988). *Tree structured fiber optics MAN's*, IEEE J.Select. Areas Comm. SAC-6, 934-943

[5] Kruskal, J .B. (1956). *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proceedings of the American Mathematical Society 7, 48-50

[6] Myung, Y.S., Lee, C.H. and Tcha, D.W. (1995). *On the Generalized Minimum Spanning Tree Problem*, Networks, VoI 26, 231-241

[7] Osman, I.H. and Laporte, G. (1996). *Metaheuristics: A Bibliography*, Annals of Operations Research, 63, 513-623
[8] Pop, P.C. (2002). *The Generalized Minimum Spanning Tree Problem*, PhD thesis, University of Twente, the Netherlands.
[9] Pop, P.C. (2004). *New models of the Generalized Minimum Spanning Tree Problem*, Journal of Mathematical Modelling and Algorithms, Volume 3, issue 2, 153-166
[10] Prim, R.C. (1957). *Shortest connection networks and some generalizations*, Bell Systems Technical Journal, 36, 1389-1401
[11] Prisco, J.J. (1986). *Fiber optic regional area networks in New York and Dallas*, IEEE J. Select. Areas Comm. SAC-4, 750-757
[12] Reeves, C.R. (1993). *Modern Metaheuristics Techniques for Combinatorial Problems*, Blackwell, Oxford.

Authors:
Petrică C. Pop - North University of Baia Mare, Baia Mare, Romania
Ioana Zelina - North University of Baia Mare, Baia Mare, Romania