Shared Multicast Trees in Ad Hoc Wireless Networks

Marika Ivanova

University of Bergen, Norway
Marika.Ivanova@uib.no
http://www.uib.no/en/persons/Marika.Ivanova

Abstract. This paper addresses a problem of shared multicast trees (SMT), which extends a recently studied problem of shared broadcast trees (SBT). In SBT, a common optimal tree for a given set of nodes allowing broadcasting from any node to the rest of the group is searched. In SMT, also nodes that neither initiate any transmission, nor act as destinations are considered. Their purpose is exclusively to relay messages between nodes. The optimization criterion is to minimize the energy consumption. The present work introduces this generalization and devises solution methods. We model the problem as an integer linear program (ILP), in order to compute the exact solution. However, the size of instances solvable by ILP is significantly limited. Therefore, we also focus on inexact methods allowing us to process larger instances. We design a fast greedy method and compare its performance with adaptations of algorithms solving related problems. Numerical experiments reveal that the presented greedy method produces trees of lower energy than alternative approaches, and the solutions are close to the optimum.

Keywords: ad hoc wireless network, Steiner tree, multicast communication, ILP model, heuristic algorithm

1 Introduction

The purpose of a multicast communication in a wireless ad-hoc network is to route information from a source to a set of destinations. Given a set of devices and distances between them, the task is to assign power to each device (node), so that the demands of the network are met and the energy consumption is as low as possible, assuming their locations are fixed. Power efficiency is an important aspect in constructing ad-hoc wireless networks since the devices are typically heavily energy-constrained. Individual devices work as transceivers, which means that they are able to both transmit and receive a signal. Moreover, the power level of a device can be dynamically adjusted during a multicast session.

Unlike wired networks, nodes in ad-hoc wireless networks use omnidirectional antennas, and hence a message reaches all nodes within the communication range of the sender. This range is determined by the power assigned to the sender, which is the maximum rather than the sum of the powers necessary to reach

all intended receivers. This feature is often referred to as the wireless multicast advantage [19].

The problems of finding power-minimizing trees in wired networks are generalizations of the minimum Steiner tree problem (e.g. [15]). Many wireless network design tasks are NP-hard [8,13]. The following problems are relevant to our work.

Minimum Energy Broadcast (MEB) is the problem of constructing an optimal arborescence for broadcasting from a given source to all remaining nodes. In order to be able to perform a broadcast session from different sources, a separate tree has to be stored for each source. A generalized multicast version assumes that the message is intended for a predefined subset of vertices. Remaining vertices can be used as intermediate nodes forwarding the message to other nodes, and possibly reduce the total cost. Such nodes are referred to as *Steiner nodes*.

Range Assignment Problem (RAP) concerns the problem of assigning transmission powers of minimum sum to the wireless nodes while ensuring network connectivity [1, 7]. Unlike MEB, the resulting links formed by the energy assignment are undirected. A generalization of the problem considers the strong connectivity within a nonempty subset of nodes.

Shared Broadcast Tree (SBT). A crucial drawback of MEB is the necessity of storing one tree for each source. The basic idea of SBT [17, 20] is to construct a common tree that is source independent and hence simplifies routing, as the relaying node does not need to know the original source in order to adjust its corresponding power level. Instead, the power level depends merely on the immediate neighbour from which the message was received. This idea is based on the observation that a signal that is being forwarded by a node does not have to reach the neighbour from which it originally came. So, if a signal comes from the most distant neighbour, the relaying power must correspond to the second most distant neighbour. When, on the other hand, a message comes from one of the closer neighbours, it has to be forwarded with the power necessary to reach the most distant one. With this conception, we get two power levels, and their selection involves only a single binary decision making.

This work introduces the shared multicast tree (SMT) problem, a generalization of SBT. Analogously to the multicast versions of MEB and RAP, in SMT we assume that there are two types of nodes, called destinations and non-destinations, respectively. Destinations can initiate a transmission, and must receive every transmission initiated by other destinations. Non-destinations can relay a message, but do not initiate any transmission. Neither do they have to receive any transmission. Passing messages via non-destinations is thus optional, and is chosen only if it saves energy, which is the main motivation for SMT. The goal is to find a common source-independent tree that connects the destinations while minimizing the power.

The decentralized nature of wireless ad-hoc networks implies its suitability for applications, where it is not possible to rely on central nodes, or where network infrastructure does not exist. This is typical for various short-term events like conferences or fixtures. Simple maintenance makes them useful in applications

such as emergency situations, disaster relief, military command and control, and most recently, in the mobile commerce sector.

2 Related Work

MEB was introduced in [19], where the authors considered three heuristic algorithms of which most cited is Broadcast Incremental Power (BIP), a greedy $\mathcal{O}(N^2 \log N)$ approximation algorithm. To our best knowledge, the most recent results for lower and upper bounds on the approximation ratio are 4.6 [3] and 6 [2], respectively. Much is written about refinements of fast sub-optimal methods, for instance [11,12,14,18]. In [7], the authors study RAP and compare cases when the resulting graph is required to be strongly and weakly connected. Several heuristic approaches are proposed (e. g. in [5,6]). Many works are also dedicated to mathematical programming techniques. Various ILP models for both MET and RAP are presented in [9, 10, 13, 16]. A special case where the transmission ranges of the stations ensure the communication between any pair of stations in at most h hops is investigated in [8].

The first work concerning SBT is [17], where the idea of a single source-independent tree embedding N broadcast trees for different sources is introduced. The authors show that using the same broadcast tree does not result in widely varying total powers for different sources. Another contribution of [17] is a polynomial-time approximation algorithm to construct a single broadcast tree, including an analysis of its performance. In [20], the authors present an ILP formulation and apply a dual decomposition method. This approach enables solving larger instances than an explicit formulation can solve, and with less than 3% performance gap to global optimality.

3 Notation and Assumptions

Let $H = (V_H, E_H)$ be an undirected graph and $u \in V_H$. The degree of u in H is denoted by $\deg_H(u)$. The input and output degree of $v \in V_K$ in a directed graph $K = (V_K, A_K)$ is denoted by $\deg^-_K(v)$ and $\deg^+_K(v)$, respectively. Let $H' = (V_{H'}, E_{H'})$ be a subgraph of H. Then, $H \setminus H'$ denotes the graph induced by the node set $V_H \setminus V_{H'}$.

A wireless network is modelled as a complete graph $G = (V, E)$, where V corresponds to the network nodes, and the edges E correspond to potential direct links between them, i.e. $\forall i, j \in V, i \neq j : \{i, j\} \in E$. The set $A = \{(i, j) : i, j \in V, i \neq j\}$ contains all arcs derived from E. Next, $D \subseteq V$ is a nonempty set of destinations with $N = |V|$ and $M = |D|$.

Let $\mathbf{z} \in \{0, 1\}^E$ be a vector with components corresponding to edges in E. The undirected graph induced by $\mathbf{z}$ is defined as $G_\mathbf{z} = (V, E_\mathbf{z})$, where $\{i, j\} \in E_\mathbf{z} \Leftrightarrow z_{ij} = 1$. The directed graph induced by $\mathbf{x} \in \{0, 1\}^A$ is defined analogously.

For $i, j \in V$, the power requirement for transmission from i to j is denoted by p_{ij}, and depends on the distance d_{ij} between i and j and environmental properties. More precisely, $p_{ij} = \kappa d^\alpha_{ij}$, where α is an environment-dependent

parameter (typically valued between 2 and 4) and κ is a constant. In this work, the power requirements p_{ij} are referred to as the *arc costs*. It follows from $d_{ij} = d_{ji}$ that the power requirements are symmetric.

If $\{i, j\}$ is an edge in a tree $T = (V_T, E_T)$, where $V_T \subseteq V$, $E_T \subseteq E$, we use $T_{i/j}$ to denote the subtree of T consisting of all vertices k such that the path from k to j visits i, as introduced in [20]. Additionally, we define a function $\mathrm{nod}(T_{i/j})$ that returns the number of destinations in $T_{i/j}$.

Neighbours of i in T are denoted i_1^T, i_2^T, $i_3^T, \ldots$ in non-increasing order of distance from i. If there is no risk of confusion, we simply omit the superscript T. The highest and second highest power levels of i are defined by its neighbours i_1 and i_2, respectively. If i is a leaf, we set $p_{ii_2} = 0$. The contribution $c_T(i)$ of i in T to the total cost depends on i's power levels:

$$c_T(i) = \mathrm{nod}(T_{i_1/i})p_{ii_2} + \mathrm{nod}(T \setminus T_{i_1/i})p_{ii_1}. \tag{1}$$

The total cost $P(T)$ of the tree T is then determined as

$$P(T) = \sum_{i \in V} c_T(i). \tag{2}$$

Problem 1. (SMT): Find a tree T in G minimizing $P(T)$ such that T spans D.

The most costly two incident edges of each node contribute to the objective function value. This reflects the nature of SBT/SMT, when the power level of a node is determined by the most costly link along which a message has to be forwarded. The power requirement of the link is multiplied by the number of senders whose transmissions are relayed through this link, which captures how often the link is used.

4 Discrete Optimization Model

We present an integer programming model for SMT, extending the model in [20] by non-destinations. In this setting we consider a set of destinations $D \subseteq V$ where a broadcast session takes place. Variables are defined as follows:

$$z_{ij} = \begin{cases} 1 & \text{if edge } \{i, j\} \in E \text{ is in } T, \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ij}^s = \begin{cases} 1 & \text{if arc } (i, j) \in A \text{ is used to transmit messages from } s \in D, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{ij}^s = \begin{cases} 1 & \text{if node } i \in V \text{ uses power } p_{ij} \text{ to transmit messages from } s \in D, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathbf{x^s} \in \{0, 1\}^A$ denote the vector consisting of variables x_{ij}^s for all $(i, j) \in A$. The ILP model is presented below. The x-variables induce $|D|$ directed trees,

that are encapsulated into a single spanning tree induced by the z-variables. The power levels are determined by the y-variables.

$$\min \sum_{(i,j)\in A} \sum_{s\in D} p_{ij} y_{ij}^s \tag{3a}$$

s.t.

$$\sum_{\{i,j\}\in E} z_{ij} \leq N-1, \tag{3b}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij}^s = 1 \qquad j,s\in D, j\neq s, \tag{3c}$$

$$x_{jk}^s \leq \sum_{i\in V\setminus\{j\}} x_{ij}^s \leq 1 \qquad j\in V\setminus D, s\in D, k\in V\setminus\{j\}, \tag{3d}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij}^s \leq \sum_{k\in V\setminus\{j\}} x_{jk}^s \qquad s\in D, j\in V\setminus D, \tag{3e}$$

$$x_{ij}^s + x_{ji}^s = z_{ij} \qquad \{i,j\}\in E, s\in D, \tag{3f}$$

$$x_{ij}^j = 0 \qquad j\in D, i\in V\setminus\{j\}, \tag{3g}$$

$$x_{ij}^s \leq \sum_{k\in V: p_{ik}\geq p_{ij}} y_{ik}^s \qquad s\in D, (i,j)\in A, \tag{3h}$$

$$\mathbf{z}\in\{0,1\}^E, \mathbf{x},\mathbf{y}\in\{0,1\}^{A\times D}. \tag{3i}$$

By constraint (3b), we express the upper bound on the number of edges in the Steiner tree. There is also a lower bound $M-1$ on the size of the spanning tree, but addition of this constraint would neither reduce the space of feasible solutions nor increase the strength of the model. If the tree does not contain any Steiner nodes, its size is the lower bound, while if all nodes are used (either as Steiner nodes, or $D=V$), its size equals the upper bound. By (3c), we ensure that a message from source s reaches a destination j from exactly one neighbour $i\in V$. Next, constraint (3d) covers the case when $j\in V\setminus D$: If a non-destination j forwards a message from s towards k, the message must come from exactly one neighbour i. Note that assuming there is no outgoing arc from a non-destination j, constraint (3d) does not prevent j from being a leaf in $G_{\mathbf{x}^s}$. We make such undesired solutions impossible by adding constraint (3e), which reduces the set of feasible solutions. However, (3e) is not necessary, because a solution, where a non-destination that does not relay any message is assigned a non-zero power, would be filtered out by the minimization procedure. The expression (3f) says that for any edge $\{i,j\}$ in $G_{\mathbf{z}}$ a message from s is transferred via either arc (i,j) or arc (j,i). The next constraint (3g) expresses that a transmission initiated by $s\in D$ cannot reach s again, which implies non-existence of a directed cycle containing s. Finally, by (3h), we define a relation between x-variables and y-variables. When arc (i,j) is used for transmission of a message from $s\in D$, vertex i relaying the message must be assigned power at least p_{ij}. The remainder of

this section justifies that the model is a correct formulation of SMT. Proofs of all claims can be found in Appendix A.

Lemma 1. *Let* $(\mathbf{x}, \mathbf{z})$ *satisfy (3b) - (3i). A replacement of all directed arcs in* $G_{\mathbf{x}^s}$ *by undirected ones yields graph* $G_{\mathbf{z}}$*, for all* $s \in D$*.*

Lemma 2. *Let* $(\mathbf{x}, \mathbf{z})$ *satisfy (3b) - (3i) and* $s \in D$*. All arcs in a path* $(s = u_1, u_2, \ldots, u_k)$ *in digraph* $G_{\mathbf{x}^s}$ *are directed from* s *towards* u_k*.*

Proposition 1. *Let* $(\mathbf{x}, \mathbf{z})$ *satisfy constraints (3b) - (3i). If* Q *is a connected component in* $G_{\mathbf{z}}$ *such that* $V_Q \cap D \neq \emptyset$*, then,* Q *does not contain any cycle.*

Proposition 2. *If* $(\mathbf{x}, \mathbf{z})$ *satisfies constraints (3b) - (3i), then there exists a path in* $G_{\mathbf{z}}$ *between any two destinations* $s, t \in D$*.*

The optimal solution to (3a) - (3i) is a graph $G_{\mathbf{z}}$ with one connected component containing all destinations. Non-destinations outside of this component are isolated vertices, as any potential links between them would be eliminated by the optimization.

Proposition 3. *If* $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ *is an optimal solution to (3a) - (3i), then one of the connected components of* $G_{\mathbf{z}}$ *is an optimal tree in Problem 1, and* $\sum_{(i,j)\in A} \sum_{s\in D} p_{ij} y^s_{ij} = P(G_{\mathbf{z}})$.

5 Inexact Methods

Solving the ILP model presented in the previous section provides the optimal power assignment, but the computation in large instances takes prohibitively long time. Hence, we now focus on algorithms with better trade-off between optimality and runtime. Any tree T in G spanning D is a feasible solution to SMT. We study the following methods:

1. Construction by MST (minimum-weight spanning tree), where all vertices are considered as destinations, and a MST is constructed over the set V.
2. Construction by BIP, where we regard the set of vertices as an instance of MEB, and apply the BIP algorithm over V.
3. Greedy Anticipating SMT algorithm described in the following Section 5.1.

The global impact of a local change suggests that the first two algorithms are rather myopic for our purposes. Unlike MST and MEB, the nature of SBT/SMT implies that an addition of an edge does not cause only a local change of power levels. Every time a new edge is appended, all nodes already included in the tree increase their contributions to the resulting cost, because the addition of an edge also increases the size of corresponding subtrees of every interior node. Therefore, the new cost cannot be calculated in constant time.

In general, the algorithms work in two phases. The first phase, construction, creates a spanning tree according to a certain strategy. Further improvements can be achieved in the second phase (refinement) which can be applied regardless of what construction method is used.

5.1 Greedy SMT Approach

We present in Alg. 1 GASMT, a greedy algorithm aimed to construct a Steiner tree $T = (V_T, E_T)$, with low SMT cost. The algorithm starts with T containing only a pre-defined root, and iteratively expands T by an edge until all destinations are present in V_T. The selection of the new edge is based on an anticipation of the entire resulting tree.

Algorithm 1 Greedy Anticipating SMT (GASMT)

Input: Complete graph $G = (V, E)$, root $r \in V$, destinations $D \subseteq V$
Output: Steiner tree $T = (V_T, E_T)$, $D \subseteq V_T \subseteq V$, $E_T \subseteq E$

```
 1: procedure BuildTree(G)
 2:     T ← ({r}, ∅)
 3:     while D ⊈ V_T do
 4:         bestCost ← ∞
 5:         Cand ← {{i, j} : i ∈ V_T, j = arg min{d_ik : k ∈ V \ V_T}}
 6:         for each {i, j} ∈ Cand do
 7:             T' ← AnticipateTree(T, i, j)
 8:             if P(T') < bestCost then
 9:                 bestCost ← P(T')
10:                 {i*, j*} ← {i, j}
11:         T ← (V_T ∪ {j*}, E_T ∪ {{i*, j*}})
        return T
12:
13: procedure AnticipateTree(T, i, j)
14:     T' ← (V_T ∪ {j}, E_T ∪ {{i, j}})
15:     Disconnected ← V \ V_T'
16:     for each v ∈ Disconnected ∩ D do
17:         u ← arg min{d_kv : k ∈ V_T'}
18:         T' ← (V_T' ∪ {v}, E_T' ∪ {{u, v}})
        return T'
```

Before a new edge is appended, we determine a set *Cand* of potential edges that can be selected: for every $u \in V_T$, we remember a potential edge linking u to the closest $v \in V \setminus V_T$ (line 5 in Alg. 1). For each candidate edge $\{i, j\}$, we build an anticipated tree spanning D. The edge $\{i, j\}$ is temporarily appended to T, which produces tree T'. Subsequently, all destinations that are not yet included in T' are connected one by one to the growing anticipated T' using the shortest possible edges. The candidate link resulting in the cheapest anticipated tree is then selected and added permanently to T. The purpose of the anticipation procedure is to predict the sizes of individual subtrees in the final tree. This allows a more realistic estimation of the resulting objective value, in contrast to the construction by MST and BIP. Non-destinations are disregarded in the anticipation procedure, because they do not alter the subtree sizes.

5.2 Refinement

Any construction algorithm can be followed by an additional phase refining the existing tree T. In particular, this phase handles non-destinations, and replaces expensive transmissions by cheaper ones.

Although the use of non-destinations may reduce the cost, the construction phase does not guarantee that all non-destinations do. Thus, cost reductions can be achieved by removing non-destinations that actually deteriorate the tree. How a non-destination v is processed depends on its degree:

- $\deg(v) = 1$: Non-destination leaves can immediately be deleted recursively.
- $\deg(v) = 2$: Let $(v_1, \ldots, v_m)$ be a maximal path in T such that $\deg(v_1) = \cdots = \deg(v_m) = 2$ and $v_1, \ldots, v_m \in V \setminus D$, and consider the two connected components T_1 and T_2 arising when the path is deleted from T. If there exists an edge $e \in E$ connecting T_1 and T_2 such that $P(T') < P(T)$, where $T' = (V_{T_1} \cup V_{T_2}, E_{T_1} \cup E_{T_2} \cup \{e\})$, the path is replaced by the best choice of e. If no such edge exists, T .
- $\deg(v) \geq 3$: Let $E(v)$ be the set of edges incident to v in T and let $T' = (V_T \setminus \{v\}, (E_T \setminus E(v)) \cup E_{\mathrm{MST}})$, where E_{MST} is the set of edges of a MST constructed over the set of v's neighbours in T. If $P(T') < P(T)$, the current tree is updated to T'.

The cost of the tree can be further improved by eliminating unnecessary transmissions by means of so called "sweep" operations [19]. After removal of an edge e, the vertices are partitioned into a cut. We then select and include the edge across the cut leading to the cheapest tree, possibly e itself. This procedure can be done for all edges, or only for selected ones - for example it makes sense to test only edges longer than a certain threshold.

6 Experimental Evaluation

We have implemented the ILP model as well as the inexact algorithms and compared numerically their performance in terms objective value and runtime.

The input parameters of the procedure generating individual instances are the number of all vertices and the number of destinations. It generates instances with the intended number of destinations and non-destinations with random coordinates uniformly distributed on a square. Finally, the power requirements are determined using $p_{ij} = \kappa d_{ij}^{\alpha}$ with $\kappa = 1$ and $\alpha = 2$. All experiments were run on an Intel Core 2 Quad CPU at 2.83 GHz and 7 GB RAM.

6.1 ILP Model

The integer programming formulation was implemented in AMPL and submitted to solver CPLEX [21] which computed optimal solutions as well as LP relaxations of the generated instances. The running time of determining the optimal solution for instances containing more than 22 vertices becomes excessively long, and so

we computed only the corresponding LP relaxations, which gives lower bounds on the objective function value.

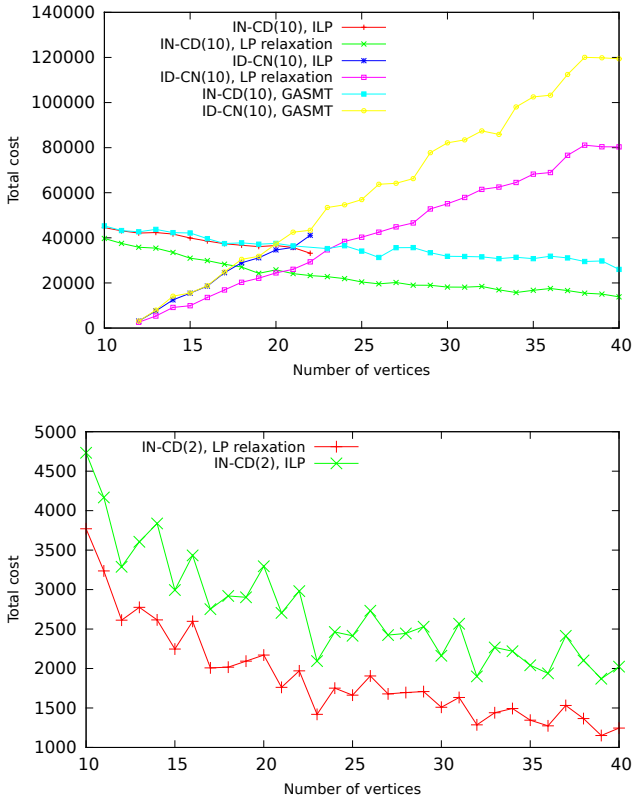

Fig. 1: Dependence of the total cost on the number of all vertices

Two instance settings were tested. In the first setting, we kept the number of non-destinations $|V \setminus D|$ constant, while increasing the number of destinations $|D|$. The abbreviation ID-CN is used to refer to this series of experiments, followed by $|V \setminus D|$ whenever it needs to be specified. In the second setting, $|D|$ was constant while $|V \setminus D|$ was increasing (IN-CD).

The first series of experiments concerns the change of the objectve function value with growing instance size. It is obvious from the graphs in Fig. 1, that in ID-CN, increasing the number of vertices also increases the total SMT cost. On the other hand, in IN-CD, the total cost decreases. This decline gradually mitigates, and it can be assumed that the average cost converges to a constant value. By way of contrast, the first graph in Fig. 1 also contains the costs obtained by the inexact algorithm. The difference between the optimum and the result of GASMT is almost negligible.

The second series of experiments shows how fast the CPU time grows with increasing number of nodes. It is apparent that the time used by the solver grows faster in ID-CN than in IN-CD, as seen in Fig. 2. Nevertheless, in both settings, the time grows exponentially. In the smallest instances, the CPU time is longer for IN-CD, because the number of destinations is higher than in ID-CN. This difference is gradually reduced. From approximately 20 vertices on, the solving time of the LP-relaxations in ID-CN becomes longer. In IN-CD(10) and ID-CN(10), every added destination causes an average increase in the ILP solution time by 89% and 208%, respectively.

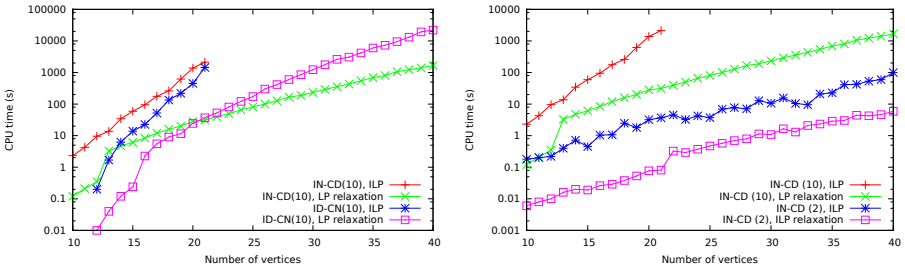

Fig. 2: Dependence of the solution time on the total number of vertices

6.2 Greedy and Heuristic Approach

The next set of experiments compares the objective value of inexact solutions produced by GASMT and the construction by MST and BIP discussed in Sect. 5. Each run of an inexact algorithm was followed by a refinement procedure, namely two iterations of non-destination removal and two iterations of sweep operations for every edge. The graphs in Fig. 3 show the results of two experimental settings, IN-CD(10) (left) and ID-CN(10) (right). Each column in the graphs corresponds to the average SMT cost calculated for 100 instances with fixed $|V|/|D|$ ratio.

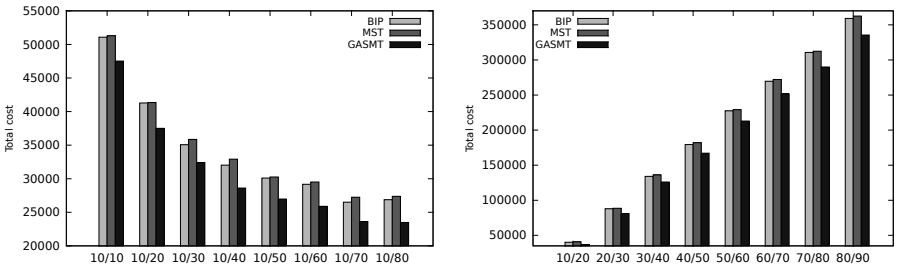

Fig. 3: Comparison of the greedy methods. The fractions on the x-axis determine the corresponding $|D|/|V|$ values.

It can be seen that the cost of the solutions produced by the construction by MST and BIP are similar, but GASMT is always perceptibly better. Nevertheless, worse time complexity of GASMT becomes apparent while processing instances of around 100 nodes, when the time spent on one instance is approximately units of minutes. On the contrary, the other two methods return the solution almost immediately.

7 Conclusion and Future Work

We have introduced a multicast version of SBT, a natural generalization of the problem. We have proposed a discrete optimization model together with the proof of its correctness. Due to the limited size of instances solvable in a practical time, heuristic and greedy approaches are also developed.

Moreover, we have conducted several numerical experiments using the CPLEX solver. It turned out that the presented ILP model can be used for solving instances with up to around 20 vertices. An increasing number of destinations causes a much faster growth of the solution time than an increasing number of non-destinations. In addition, these experiments give us insight into the cost reduction as a function of increasing number of non-destinations.

Further experiments involved the inexact methods. The GASMT algorithm presented in this work gives better results than the construction by MST and BIP applied on SMT. Moreover, the solutions provided by GASMT are close to the optimal ones determined by solving the ILP model.

A subject of continued research is a detailed theoretical study of the inexact methods. There are several interesting questions regarding this topic, like whether any inexact method is an approximation algorithm for SBT/SMT, or whether any method performs consistently better than others. There is also a substantial room for further improvements of the ILP model so that it can be applicable for larger instances. In particular, methods like strong valid inequalities, lazy constraints and user cuts could serve for this purpose.

References

1. Althaus, E., Calinescu, G., Mandoiu, I.I., Prasad, S., Tchervenski, N., Zelikovsky, A.: Power efficient range assignment in ad-hoc wireless networks. Wireless Communications and Networking. 3, 1889–1894 (2003)
2. Ambühl, C.: An Optimal Bound for the MST Algorithm to Compute Energy Efficient Broadcast Trees in Wireless Networks. ICALP'05 Proceedings of the 32nd international conference on Automata, Languages and Programming. 1139–1150 (2005)
3. Bauer, J., Haugland, D., Yuan, D.: New results on the time complexity and approximation ratio of the Broadcast Incremental Power algorithm. INFORMATION PROCESSING LETTERS. 109, 12, 615–619 (2009)
4. Bauer, J., Haugland, D., Yuan, D.: A fast local search method for minimum energy broadcast in wireless ad hoc networks. Operations Research Letters. 37, 2, 75–79 (2009)

5. Bein, D., Zheng, S. Q.: Energy efficient all-to-all broadcast in all-wireless networks. Information Sciences. 180, 10, 1781–1792 (2010)
6. Bhukya, W.N., Singh, A.: An effective heuristic for construction of all-to-all minimum power broadcast trees in wireless networks. Advances in Computing, Communications and Informatics. 74–79 (2014)
7. Blough, D.M., Leoncini, M., Resta, G., Santi, P.: On The Symmetric Range Assignment Problem In Wireless Ad Hoc Networks. Proceedings of the IFIP 17th World Computer Congress - TC1 Stream / 2Nd IFIP International Conference on Theoretical Computer Science: Foundations of Information Technology in the Era of Networking and Mobile Computing. 71–82 (2002)
8. Clementi, E.F., Penna, P., Silvestri, R.: On the power assignment problem in radio networks Mobile Networks and Applications - Discrete algorithms and methods for mobile computing and communications. 9, 2, 125–140 (2004)
9. Das, A. K., Marks, R. J., El-sharkawi, M.: Minimum power broadcast trees for wireless networks: Integer programming formulations. The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies. 245–248 (2003)
10. Das, A.K.; Marks, R.J., II; El-Sharkawi, M.; Arabshahi, P.; Gray, A.: Optimization methods for minimum power bidirectional topology construction in wireless networks with sectored antennas. Global Telecommunications Conference. 6, 3962–3968 (2004)
11. Das, A.K., Marks, R.J., El-sharkawi, M., Arabshahi, P., Gray, A.: r-Shrink: A heuristic for improving minimum power broadcast trees in wireless networks. Proceedings of the IEEE GLOBECOM'03. 1, 523–527 (2003)
12. Das, A.K., Marks, R.J., El-sharkawi, M., Arabshahi, P., Gray, A.: e-Merge : A heuristic for improving minimum power broadcast trees in wireless networks Technical Report, Department of Electrical Engineering, University of Washington. 2003.
13. Haugland, D., Yuan, D.: Wireless Network Design: Optimization Models and Solution Procedures. International Series in Operations Research & Management Science. New York, Springer. 219–246 (2011)
14. Liang, W.: Constructing Minimum-energy Broadcast Trees in Wireless Ad Hoc Networks. Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing. 112–122 (2002)
15. Hwang, F.K., Richards, D.S., Winter, P.: The Steiner Tree Problem. Annals of Discrete Mathematics. Elsevier Science. (1992)
16. Montemanni, R., Gambardella, L.M.: Exact Algorithms for the Minimum Power Symmetric Connectivity Problem in Wireless Networks. Computers and Operations Research. 32, 11, 2891–2904 (2005)
17. Papadimitriou, I., and Georgiadis, L.: Minimum-energy Broadcasting in Multi-hop Wireless Networks Using a Single Broadcast Tree. Mobile Networks and Applications. 11, 3, 361–375 (2006)
18. Wan, P., Clinescu, G., Yi C.: Minimum-Power Multicast Routing in Static Ad Hoc Wireless Networks. IEEE/ACM Transactions on Networking (TON). 12, 3, 507–514 (2004)
19. Wieselthier, J. E., Nguyen, G. D., Ephremides, A.: On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. 2, 585–594 (2000)
20. Yuan, D., Haugland, D.: Dual Decomposition for Computational Optimization of Minimum-Power Shared Broadcast Tree in Wireless Networks. IEEE Transactions on Mobile Computing. 12, 11, 2008–2019 (2012)
21. IBM ILOG CPLEX V12.1 User's Manual for CPLEX. 2009

Appendix A

Lemma 1. *Let* $(\mathbf{x}, \mathbf{z})$ *satisfy (3b) - (3i). A replacement of all directed arcs in* $G_{\mathbf{x}^s}$ *by undirected ones yields graph* $G_{\mathbf{z}}$*, for all* $s \in D$*.*

Proof. By constraint (3f), $\{i, j\} \in E_{\mathbf{z}}$ if and only if $(i, j) \in E_{\mathbf{x}^s}$ or $(j, i) \in E_{\mathbf{x}^s}$. The lemma then follows immediately. □

Lemma 2. *Let* $(\mathbf{x}, \mathbf{z})$ *satisfy (3b) - (3i) and* $s \in D$*. All arcs in a path* $(s = u_1, u_2, \ldots, u_k)$ *in digraph* $G_{\mathbf{x}^s}$ *are directed from* s *towards* u_k*.*

Proof. By (3g), we have $x^s_{u_2,s} = 0$, hence $x^s_{s,u_2} = 1$. Let us assume that there is a directed path from s to u_i. Then $x^s_{u_i,u_{i+1}} = 1$ holds, because if $x^s_{u_{i+1},u_i} = 1$, constraint (3c) would be violated (in case $u_i \in D$) or constraint (3d) would be violated (in case $u_i \in V \setminus D$), since we already assumed that $x^s_{u_{i-1},u_i} = 1$. In other words, the constraints (3c) - (3d) ensure that for any vertex there is no more than 1 inbound arc with x^s-value 1. The result then follows by induction on i. □

Proposition 1. *Let* $(\mathbf{x}, \mathbf{z})$ *satisfy constraints (3b) - (3i). If* Q *is a connected component in* $G_{\mathbf{z}}$ *such that* $V_Q \cap D \neq \emptyset$*, then,* Q *does not contain any cycle.*

Proof. By contradiction, let us assume that there exists a cycle $(c_1, c_2, \ldots, c_p = c_1)$ in Q. From Lemma 1, we have a corresponding directed subgraph C in $G_{\mathbf{x}^s}$ for any $s \in D$. Depending on the arc orientations, C is either a directed cycle or a directed acyclic graph.

If C is not a directed cycle, it contains at least one vertex c_i with $\deg^C_-(c_i) = 2$. This means that $x^s_{c_{i-1},c_i} = x^s_{c_{i+1},c_i} = 1$, which violates constraint (3c) if $c_i \in D$ or (3d) if $c_i \in V \setminus D$. Therefore, C is a directed cycle. As every $c_i \in C$ has $\deg^C_-(c_i) = \deg^C_+(c_i) = 1$, we have $x^s_{c_{i-1},c_i} = x^s_{c_i,c_{i+1}} = 1$, for all $s \in D$. Thus, C can not contain $s \in D$, because that would contradict constraint (3g), which says that the input degree of $s \in D$ in $G_{\mathbf{x}^s}$ is zero.

The remaining possibility is that C is a directed cycle containing only vertices in $V \setminus D$. As Q contains at least one destination $s \in D$, there is a path $(s = u_1, u_2, \ldots u_l)$ connecting s and C. Without loss of generality, let $c_1 = u_l$ be the vertex in C closest to s. By Lemma 2, we see that all edges of the path from s to c_1 in $G_{\mathbf{x}^s}$ are directed from s, which means that also $x^s_{u_{l-1},c_1} = 1$. However, that contradicts (3d), because we already assumed $x^s_{c_{p-1},c_1} = 1$ and $u_{l-1} \neq c_{p-1}$. It follows that the subgraph C is neither a cyclic graph nor a directed acyclic graph, which is a contradiction completing the proof. □

Proposition 2. *If* $(\mathbf{x}, \mathbf{z})$ *satisfies constraints (3b) - (3i), then there exists a path in* $G_{\mathbf{z}}$ *between any two destinations* $s, t \in D$*.*

Proof. Because (3g) implies that $\deg^{G_{\mathbf{x}^s}}_-(s) = 0$, the claim is equivalent to the following: Any maximal directed path in $G_{\mathbf{x}^s}$ to t starts in s. Assume the contrary, that the path starts in $j \neq s$. If $j \in D$, constraint (3c) ensures that there is one inbound arc (i, j), and hence the path can be extended by (i, j). Similarly, if

$j \in V \setminus D$, by (3d), if j has an outgoing arc (which is fulfilled by the assumption), there must be an inbound arc, which again contradicts the maximality of the path. According to Proposition 1, $G_{\mathbf{z}}$ is acyclic, and the proof is complete since the number of vertices in V is finite. □

Proposition 3. *If* $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ *is an optimal solution to (3a) - (3i), then one of the connected components of* $G_{\mathbf{z}}$ *is an optimal tree in Problem 1, and* $\sum_{(i,j)\in A} \sum_{s\in D} p_{ij} y_{ij}^s = P(G_{\mathbf{z}})$.

Proof. It follows from Prop. 2 that $G_{\mathbf{z}}$ has a connected component $d(G_{\mathbf{z}})$ spanning D, and from Prop. 1 that $d(G_{\mathbf{z}})$ is a tree. Also, it is obvious that for any tree T in G spanning D, there exists a feasible solution $(\mathbf{x}', \mathbf{y}', \mathbf{z}')$ to (3a) - (3i) such that $d(G_{\mathbf{z}'}) = T$. Consequently, we only need to show that for all $i \in V$,

$$\sum_{j\in V:(i,j)\in A} \sum_{s\in D} p_{ij} y_{ij}^s = c_{G_{\mathbf{z}}}(i) \tag{4}$$

as defined by (1).

Consider an arbitrary vertex i. If i is a leaf in $d(G_{\mathbf{z}})$, it must be a destination. Recall that i_1 denotes the most distant, in this case the only, neighbour of i in $G_{\mathbf{z}}$. Because of (3f) and (3g), we have $x_{ii_1}^i = 1$. Constraint (3h) then enforces $y_{ii_1}^i = 1$, which implies (4).

If i is an interior vertex, constraints (3c) and (3d) enforce that there is one inbound arc to i in $G_{\mathbf{x}^s}$. So, either $x_{i_1 i}^s = x_{ii_2}^s = 1$ or $x_{i_2 i}^s = x_{ii_1}^s = 1$. In the first case, (3h) assigns $y_{ii_2}^s = 1$. In the second case, $y_{ii_1}^s = 1$ holds. Due to minimization, all variables $y_{ii_j}^s$ with $j > 2$ are set to 0. As the first case corresponds to $s \in T_{i_1/i}$ and the second case to $s \in T \setminus T_{i_1/i}$, summing over all $s \in D$ yields (4). The contribution of i to (2) is then $nod(T_{i_1/i}) p_{ii_2} + nod(T \setminus T_{i_1/i}) p_{ii_1}$. □

By satisfying the constraints, we obtain a solution where each vertex is assigned powers necessary for relaying a message from a particular source. By optimality, the solution involves the power levels representing minimal SMT cost.