Open access • Journal Article • DOI:10.1109/TMC.2011.231

Dual Decomposition for Computational Optimization of Minimum-Power Shared Broadcast Tree in Wireless Networks — Source link

Di Yuan, Dag Haugland

Institutions: Linköping University, University of Bergen

Published on: 01 Dec 2012 - IEEE Transactions on Mobile Computing (IEEE)

Topics: Tree (data structure), Decomposition method (constraint satisfaction), Broadcast communication network, Assignment problem and Local search (optimization)

Dual Decomposition for Computational Optimization of Minimum-Power Shared Broadcast Tree in Wireless Networks

Di Yuan and Dag Haugland

Di Yuan and Dag Haugland, Dual Decomposition for Computational Optimization of Minimum-Power Shared Broadcast Tree in Wireless Networks, 2012, IEEE Transactions on Mobile Computing, (11), 12, 2008-2019.
http://dx.doi.org/10.1109/TMC.2011.231

Dual Decomposition for Computational Optimization of Minimum-Power Shared Broadcast Tree in Wireless Networks

Di Yuan*

Department of Science and Technology, Linköping University,
SE-601 74 Norrköping, Sweden
Email: diyua@itn.liu.se

Dag Haugland

Department of Informatics, University of Bergen
PB. 7800, N-5020 Bergen, Norway
Email: Dag.Haugland@ii.uib.no

Abstract

We consider the problem of constructing a shared broadcast tree (SBT) in wireless networks, such that the total power required for supporting broadcast initiated by all source nodes is minimal. In the well-studied minimum-energy broadcast (MEB) problem, the optimal tree varies by source. In contrast, SBT is source-independent, thus substantially reducing the overhead for information storage and processing. The SBT problem also differs from the range assignment problem (RAP), because the power for message forwarding in SBT, although being source-independent, depends on from which tree neighbor the message is received. We approach SBT from a computational optimization standpoint, and present a dual decomposition method applied to an optimization model that embeds multiple directed trees into a shared tree. For the dual decomposition method, some of the constraints in the model are preferably formulated implicitly. The dual decomposition scheme is coupled with a fast local search algorithm. We report computational results demonstrating the effectiveness of the proposed approach. In average, the performance gap to global optimality is less than three percent.

Index Terms: shared broadcast tree, wireless networks, discrete optimization, dual decomposition.

*Corresponding author.

1 Introduction

Energy efficiency in infrastructureless wireless networks has attracted a great amount of research attention. The issue is of particular relevance to wireless ad hoc networks, where communication devices are typically heavily energy-constrained. One optimization task in this context is to perform broadcast using a minimum amount of transmission power. Broadcast communications are used not only for data of broadcast nature, but also frequently for disseminating control information. In this paper, we consider the problem of constructing a power-optimal shared broadcast tree (SBT) that is used by broadcast sessions initiated by all source nodes.

For power-optimal broadcast in wireless networks, a majority of the literature has considered two problem types that are related to but different from SBT: the minimum-energy broadcast (MEB) problem [1], and the range assignment problem (RAP, e.g., [2]). MEB amounts to constructing a minimum-power directed tree, i.e., an arborescence, to be used for broadcast from a given source node. The optimal tree is source-specific. Consequently, the power level to use is dependent on message source. Consider the network scenario of N nodes that all are potential sources of broadcast sessions. Applying MEB to the scenario, N different trees, one per source, need to be computed and stored. Upon receiving a broadcast message, a relaying node has to open the message up to the level necessary to retrieve the source, in order to determine which of the N MEB trees to apply and the corresponding power level. In contrast, broadcast by SBT differs from MEB in two major aspects. First, only a single tree needs to be stored. Second, there is no need of retrieving the message's original source in relaying. In SBT, the power that a node uses to forward a broadcast message is only dependent on the previous one-hop tree neighbor from which the message is received. In fact, as will be detailed in Section 2, determining the power for relaying a message involves only a single binary yes-or-no decision making. Although for an individual source node, the broadcast power by the optimal SBT may be higher than that of the optimal MEB tree, the use of SBT is justified by the substantially reduced amount of information storage and message processing.

SBT differs also from RAP. In the latter, a power corresponding to a radio range is chosen for each node to provide a strongly connected topology by bi-directional links, with the objective of minimizing the total power assignment. In an SBT, the power used by a node to relay a broadcast message is the one required to reach all its tree neighbors except the neighbor from which the message is received. Hence, the node does make a choice on the power level in each forwarding operation, although the choice is independent of the original message source. This makes the structure of SBT very different from that of RAP.

Similar to MEB, the study of SBT is motivated by optimization aspects of energy-efficient dissemination of broadcast information in ad hoc networks. Potential applications of the networking concept can be found in military communications, disaster relief, and sensor networking (see, e.g, [3, 4, 5, 6], for details). When all nodes are potential broadcast sources, SBT provides an alternative, and, from an information storage and processing standpoint, more elegant approach to MEB, of which the importance has attracted a large amount of research efforts (see [1, 7, 8, 9, 10] and Section 2). We also remark that the contribution of the paper targets the underlying performance optimization principles and foundation rather than designing a specific system protocol; to this end, the paper focuses on algorithmic and computational aspects of the SBT optimization problem.

The authors of [11] introduced the SBT problem, and presented results on complexity and approximation algorithm. To the best of our knowledge, formulating and solving the SBT problem via discrete optimization has remained open. Discrete optimization potentially enables to numerically approach exact optimum. In addition and more importantly, an effective discrete optimization approach typically yields a numerically tight interval confining the optimum, when exact optimum becomes beyond reach due to problem size. In both cases, the results are very useful for assessing the performance of fast but heuristic algorithms, including distributed ones.

In this paper, we present a mathematical programming model for power-optimal SBT. The model embeds multiple directed trees, one per source node, into a shared tree. A feature of the model is that the relationship between the directed trees and the power levels is preferably kept implicit, even if it can be formulated by explicit equations. This is because we develop a dual decomposition method, such that the resulting Lagrangean subproblem can be solved exactly in polynomial time without requiring an explicit mathematical representation of how power levels are set in the directed trees. We use subgradient optimization to approach the Lagrangean dual optimum. The dual decomposition method is coupled with a local search algorithm that starts from the solution of the Lagrangean subproblem and searches for good SBT solutions. We report computational experiments for networks defined over complete graphs of up to 80 nodes. The proposed method is highly effective in delivering solutions being very close to optimum. In average, the performance gap to global optimality is below three percent.

The remainder of the paper is organized as follows. In Section 2 we formalize the SBT problem, compare SBT to MEB and RAP using an illustrative example, and review some related works. The mathematical programming model is presented in Section 3. We discuss the dual decomposition method and the local search algorithm in Sections 4 and 5, respectively. Computational results are reported in Section 6, followed by conclusions in Section 7.

2 Preliminaries

Consider a wireless network modeled by graph $G = (V, E)$, where V and E denote the sets of nodes (network units) and potential links, respectively. Let $N = |V|$. The elements in E are undirected edges. Without any loss of generality, we assume $i < j$ for all $(i, j) \in E$. We use A to denote the set of arcs derived from E, i.e., $A = \{(i, j) : i, j \in V, (i, j)$ or $(j, i) \in E\}$. Notation $i \in G$ and $(i, j) \in G$ imply respectively that i is a node and that (i, j) is an edge in G. A similar notation is adopted for trees in G. For any tree T, we let $|T|$ denote the number of nodes in T, and for any edge $(i, j) \in T$, $T_{i/j}$ denotes the subtree containing node i when (i, j) is deleted from T. The power parameter of $(i, j) \in A$ is denoted by p_{ij}. Due to the wireless multicast advantage [1], the power level p_{ij} allows node i to reach all nodes in the set $\{k \in V : p_{ik} \leq p_{ij}\}$. There is a widely used formula for setting the power parameters in the literature. The formula is not assumed in our optimization model or the solution method, even so it is used for generating test networks. In this power formula, $p_{ij} = \kappa d_{ij}^{\alpha}$, where d_{ij} is the distance between i and j, κ is a constant, and α is between two and four.

The task of MEB is to make a minimum-sum power assignment to create connectivity from a source node s to $V \setminus \{s\}$. The optimum is characterized by a directed tree, i.e., an arborescence. Denoting by S_i the children of i in s-rooted arborescence T^s, the power of node i to forward a broadcast message of s equals the maximum of the power required to reach all children in S_i, or zero if i is a leaf. Solving MEB amounts to constructing T^s minimizing $\sum_{i \in V} \max_{j \in S_i} p_{ij}$.

An SBT, on the other hand, is an undirected spanning tree T. For each source node $s \in V$, T maps uniquely into an arborescence T^s rooted at s, giving the nodes' broadcast power levels for source s in the shared tree T. An SBT is optimal if the total broadcast power for all source nodes, or equivalently, the average broadcast power per source, is minimum among all spanning trees.

For a formal definition of the total broadcast power of an SBT, suppose that the highest and second highest powers of node i are defined by its tree neighbors j_1 and j_2, respectively. Clearly, j_1 and j_2 depend on T and i, but for reasons of notational simplicity, we do not introduce notation reflecting this dependency. For all source nodes reaching i via j_1, i.e., nodes in $T_{j_1/i}$, node i uses power p_{ij_2} for forwarding their messages, whereas for all the remaining nodes including i itself, the power equals p_{ij_1}. If i is a leaf, node i spends power p_{ij_1} only for its own broadcast session. Node j_2 is in this case undefined, but for notational convenience we let $p_{ij_2} = 0$. The decision of selecting the power level at node i is simply made by observing whether or not the message is received from j_1. If it is, the answer is p_{ij_2}, while power p_{ij_1} is required otherwise. Knowledge to the message source is superfluous in this respect. The power-optimal SBT is formalized as follows:

[SBT] Find a spanning tree T of G minimizing $\displaystyle\sum_{i \in V} \big(|T_{j_1/i}| p_{ij_2} + (N - |T_{j_1/i}|) p_{ij_1}\big)$.

The objective function of SBT differs from that of RAP, as the former accounts for how many source nodes use each of the two power levels. RAP, sometimes also referred to as the minimum power symmetric connectivity problem [12], originates from topology control in wireless networks [13]. The objective is to find a power allocation yielding strong connectivity between all nodes by bi-directional links, which is equivalent to constructing a spanning tree T with minimum $\sum_{i \in V} p_{ij_1}$.

Both MEB and RAP have been well-studied in the literature. MEB was introduced by Wieselthier et al. [1, 14]. The authors proposed several solution algorithms. Among them, the most cited one is the broadcast incremental power (BIP) algorithm that is an adaptation of Prim's MST algorithm to MEB. The NP-hardness of MEB was established in [8], and the MEB literature is rich on studies of approximation algorithms [10, 15, 16, 17, 18, 19, 20, 21] and heuristic algorithms [19, 22, 23, 24, 25, 26, 27]. Integer programming formulations were first presented in [7], and more recently in [27, 28, 29]. The authors of [4] presented integer programming and dynamic programming techniques for MEB with a hop limit. For minimum-energy multicast, research results are provided in [1, 9, 14, 26, 27, 28].

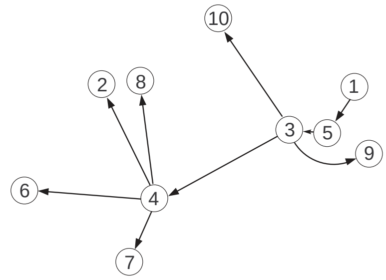
For RAP, NP-hardness results were provided in [30, 31]. For the special case of 1-dimensional Euclidean space, RAP admits polynomial time algorithms [32, 33]. Integer programming and approximation schemes have been investigated in [2]. Further developments of integer programming formulations and approaches have been presented [12, 34]. In [35], the authors considered using the RAP solution for all-to-all communications, and compared several heuristic algorithms. An extension of RAP is to have a hop limit on the tree path connecting node pairs, i.e., diameter-bounded RAP [36, 37, 38]. Another extension of RAP is to require that the range assignment is large enough to handle node failure [39].

In comparison to MEB and RAP, research results on computing optimal SBT are available to a much lesser extent. Papadimitriou and Georgiadis [11] proved the NP-hardness of SBT, and presented numerical experiments with an approximation algorithm. They also proved that, in an SBT, the difference in the broadcast power consumption for any two sources is bounded by a factor of two, i.e., SBT exhibits a fairness guarantee among the source nodes. This fact justifies further the relevance of SBT to broadcast communications, in addition to the aforementioned advantages in terms of small storage requirement and simplified message processing.

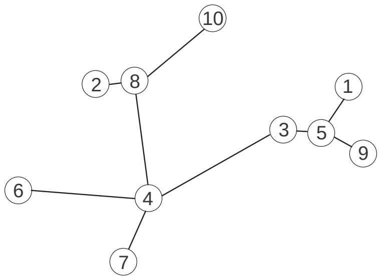
In Figure 1, we give an illustrative example of a small network of 10 nodes. The power parameter matrix is symmetric, and shown in Figure 1(a). The optimum solutions of MEB for one source node, RAP, and SBT are illustrated in Figures 1(b), 1(c), and 1(d), respectively.

	1	2	3	4	5	6	7	8	9	10
1	-	103.08	12.00	87.12	6.42	217.10	145.65	84.20	7.70	44.14
2		-	55.88	26.63	76.70	32.37	60.44	1.00	118.87	33.93
3			-	34.51	1.70	131.40	74.70	43.46	11.75	37.09
4				-	9.11	37.92	10.54	25.43	75.34	75.38
5					-	162.02	91.96	61.77	4.69	46.61
6						-	37.71	42.87	213.99	132.11
7							-	62.84	118.32	142.03
8								-	100.31	24.51
9									-	76.08
10										-

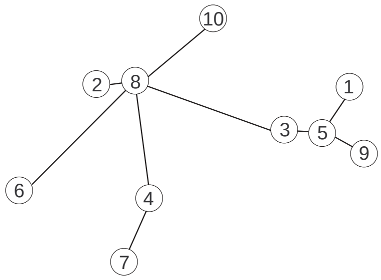
(a) Power matrix of the network instance.



(b) Optimal MEB tree for source node one.



(c) Optimal RAP tree.



(d) Optimal SBT.

Figure 1: An illustrative example of various power-optimal trees.

One can see that the three optimal trees all differ from each other. The broadcast power of the optimal MEB tree for source node one is 83.12. When SBT is used, the power for the same source is 107.97. If we compute the optimum of MEB for the remaining source nodes (not shown in the figure), the average MEB power over all source nodes is 77.00. Using SBT, the average power per source is 90.39. Thus the difference between SBT and optimal source-specific broadcast is about 17% for this example. In RAP, the highest power level of each node is counted exactly once, hence the tree in Figure 1(c) is better than the SBT optimal tree in Figure 1(d), because the highest power levels at nodes 4, 6, and 8 are lower in the former. However, the optimum RAP tree is not optimal when the number of sources using each power level is taken into account. For source nodes 2, 8, and 10, the RAP tree requires one additional forwarding by node 4 in comparison to the SBT tree. In the optimum of SBT, this hop is avoided.

3 A Discrete Optimization Model

We present a discrete optimization model for SBT. The model uses three sets of binary variables.

$$z_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \in E \text{ is in the SBT,} \\ 0 & \text{otherwise.} \end{cases}$$

$$q_{ij}^t = \begin{cases} 1 & \text{if arc } (i,j) \in A \text{ is used to reach node } t \in V, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ij}^s = \begin{cases} 1 & \text{if node } i \in V \text{ uses power } p_{ij} \text{ to broadcast the message of source } s \in V, \\ 0 & \text{otherwise.} \end{cases}$$

The model is presented below. The explicit constraints in the model embed multiple directed trees, represented by the q-variables, into a spanning tree represented by the z-variables. Note that the relation between $\mathbf{q}$ and $\mathbf{y}$ is modeled implicitly.

$$\min \quad \sum_{(i,j)\in A} \sum_{s\in V} p_{ij} y_{ij}^s \qquad (1)$$

$$\text{s.t.} \quad \sum_{(i,j)\in E} z_{ij} = N - 1 \qquad (2)$$

$$\sum_{j\in V:(i,j)\in A} q_{ij}^t = 1 \qquad i \in V, t \in V : i \neq t \qquad (3)$$

$$q_{ij}^t + q_{ji}^t = z_{ij} \qquad (i,j) \in E, t \in V \qquad (4)$$

$$q_{ij}^i = 0 \qquad (i,j) \in A \qquad (5)$$

$$\mathbf{y} = Q(\mathbf{q}) \qquad (6)$$

$$\mathbf{z} \in \{0,1\}^{|E|} \mathbf{y}, \mathbf{q} \in \{0,1\}^{|A|\times|V|} \qquad (7)$$

The objective function (1) minimizes the total broadcast power for all source nodes. By (2), exactly $N-1$ edges must be selected; this is a necessary condition for a spanning tree. Equations (3) ensure that, at any node $i \in V$, exactly one arc is selected to reach each node

$t \in V \setminus \{i\}$. Equations (4) express the relation between the z- and q-variables in a tree: For any tree edge (i, j), every node $t \in V$ is reached either via arc (i, j) or arc (j, i). If edge (i, j) is not present in the tree, then clearly all q-variables must be zeros on this edge. The next set of constraints (5) forbids any arc to be used to reach the head node. Finally, in (6) an implicit function, denoted by Q, models the power levels induced by the q-variables.

Let $G_{\mathbf{z}}$ denote the graph induced by $\mathbf{z} \in \{0,1\}^{|E|}$. To show that (2)-(7) give a valid formulation of SBT, we first prove that $G_{\mathbf{z}}$ is a spanning tree for any solution to the formulation.

Proposition 1 *Assume* $(\mathbf{z}, \mathbf{q})$ *satisfies equations* (2)–(5) *and* (7). *Then 1)* $G_{\mathbf{z}}$ *is a spanning tree, and 2) for each* $t \in V$ *and each* $(i, j) \in G_{\mathbf{z}}$*,* $q^t_{ij} = 1$ *(*$q^t_{ji} = 1$*) if the direction toward* t *on edge* (i, j) *is from* i *to* j *(from* j *to* i*).*

Proof Because of (2), 1) holds if $G_{\mathbf{z}}$ has no cycles. Assume there is a cycle containing node t. For convenience and without loss of generality, assume the cycle consists in $t, 1, \ldots, n, t$, with the convention that $n + 1 = t$. The sequence $1, \ldots, n, t$ forms an elementary path. From $z_{nt} = 1$ and (4)–(5), it follows that $q^t_{n,n+1} = 1$. Let $k = n$. Since $q^t_{k,k+1} = 1$ and $k \neq t$, (3) implies $q^t_{k,k-1} = 0$. In addition, because $z_{k-1,k} = 1$, (4) yields $q^t_{k-1,k} = 1$. By induction in k, we hence obtain $q^t_{k,k+1} = 1, k = 1, \ldots, n$. Applying the same argument to path $n, \ldots, 1, t$, we get $q^t_{12} = \cdots = q^t_{nt} = q^t_{n,n-1} \cdots = q^t_{1t} = 1$, contradicting (4). Therefore property 1) holds. Since $G_{\mathbf{z}}$ is a spanning tree, Property 2) follows immediately from the proven fact that any elementary path $1, \ldots, t$ in $G_{\mathbf{z}}$ has $q^t_{k,k+1} = 1$ for $1 \leq k \leq n$. ∎

By the proposition, the model embeds N directed trees into an undirected, shared tree. The values of $\mathbf{q}$ carry the information of the power levels of every node for reaching its tree neighbors. Following the discussion in Section 2, the function Q maps $\mathbf{q}$ to a vector $\mathbf{y}$, in which the component corresponding to node $i \in V$ is given according to the following two rules. For both cases, all y-variables that are not set to be one are zeros.

Case 1: There is a single outgoing arc, say (i, j_1), with q-variables with value one. That is, $q^t_{ij_1} = 1, \forall t \in V, t \neq i$, and $q^t_{ij} = 0, \forall j \neq j_1, t \in V$. In this case, $y^i_{ij_1} = 1$.

Case 2: There are two or more outgoing arcs, for which the q-variables equal one. Among them, let (i, j_1) and (i, j_2) denote the two having highest and second highest power levels, respectively. Note that, by (4)–(5), $q_{ij_1}^{j_1} = 1$ and $q_{ij_2}^{j_2} = 1$ (and $q_{ij_1}^{i} = q_{ij_2}^{i} = 0$). For $s \in V$, function Q sets $y_{ij_1}^{s} = 1$ if $q_{ij_1}^{s} = 0$, otherwise $y_{ij_2}^{s} = 1$. In the former case, s is not reached via (i, j_1), thus forwarding the message of s has to use p_{ij_1}. The latter case means that using power p_{ij_2} is sufficient since the message of s arrives i via (j_1, i).

The above definition of Q, together with Proposition 1, establish the correctness of the model for SBT. We remark that there are several ways to formulate the function Q explicitly. The simplest choice is to use the following set of inequalities.

$$q_{ji}^{s} \leq \sum_{k: p_{ik} \geq p_{ij}} y_{ik}^{s}, i \in V, s \in V. \tag{8}$$

The inequality says that if (j, i) is used to reach s, i.e., $q_{ji}^{s} = 1$, then the power used by node i to relay broadcast message of s must be at least p_{ij}. Note that the inequality does not explicitly set any y-variable to zero, though it is straightforward to verify that, at optimum, the effect of (8) is identical to the definition of Q. From a computational standpoint, solving the formulation with (8) does not lead to an efficient solution method, see Section 6. Our dual decomposition algorithm can effectively consider Q without any explicit equations. In fact, no matter which explicit formulation of Q is used, the continuous relaxation will never be able to produce a better estimation on the global optimum than the dual optimum to which our algorithm converges.

4 Dual Decomposition

We dualize (4) and denote by $\mu_{ij}^{t}, (i, j) \in E, t \in V$ the corresponding Lagrangean multipliers. These multipliers are defined for the edges, but the effect on the two end nodes of each edge will differ. In the subsequent text, we use $\mu_{[ij]}^{t}$ to denote μ_{ij}^{t} if $i < j$, and μ_{ji}^{t} otherwise. Note that the multipliers are not restricted in sign. After dualizing (4), the term

$\sum_{(i,j)\in A}\sum_{t\in V}\mu^t_{[ij]}q^t_{ij} - \sum_{(i,j)\in E}(\sum_{t\in V}\mu^t_{ij})z_{ij}$ is added to the objective function (1). The Lagrangean subproblem decomposes into two parts. The first part, involving the z-variables, becomes the following optimization problem.

$$
\begin{aligned}
\text{[S1]}\quad L_{\mathbf{z}}(\boldsymbol{\mu}) = \min & \sum_{(i,j)\in E}(-\sum_{t\in V}\mu^t_{ij})z_{ij} \\
\text{s.t.} & \sum_{(i,j)\in E} z_{ij} = N-1 \\
& \mathbf{z} \in \{0,1\}^{|E|}
\end{aligned}
\tag{9}
$$

Solving subproblem S1 to optimality means to select the $N-1$ edges with the lowest cost values, where the cost of $(i,j)\in E$ equals $-\sum_{t\in V}\mu^t_{ij}$. However, to be qualified as a feasible solution to SBT, $\mathbf{z}$ shall form a spanning tree. Hence we strengthen the solution by adding the spanning-tree requirement, with which S1 becomes, in fact, a minimum spanning tree (MST) problem. Not only does this MST solve S1, it also provides a candidate SBT. Thus after solving S1, our decomposition method evaluates the tree's broadcast power for all source nodes.

The second subproblem amounts to optimizing the values of the q- and y-variables. This subproblem is formulated below.

$$
\begin{aligned}
\text{[S2]}\quad L_{\mathbf{q},\mathbf{y}}(\boldsymbol{\mu}) = \min & \sum_{(i,j)\in A}\sum_{s\in V}p_{ij}y^s_{ij} + \sum_{(i,j)\in A}\sum_{t\in V}\mu^t_{[ij]}q^t_{ij} && \\
\text{s.t.} & \sum_{j\in V:(i,j)\in A} q^t_{ij} = 1 && i\in V, t\in V : i\neq t \\
& q^i_{ij} = 0 && (i,j)\in A \\
& \mathbf{y} = Q(\mathbf{q}) && \\
& \mathbf{y},\mathbf{q} \in \{0,1\}^{|A|\times|V|} &&
\end{aligned}
\tag{10}
$$

From the problem structure and the definition of Q, it follows that S2 decomposes further by node, resulting in N separate problems. For each node i, exactly one outgoing arc must be chosen for each destination $t \neq i$. The choice is made to minimize the cost, under the condition that it has to be consistent with the power levels of i. A polynomial-time enumeration scheme is able to deliver the optimum to S2. The enumeration goes through the following cases spanning all relevant power-setting scenarios of node i, and selects the one giving minimum total cost.

Case 1: Node i has a single power level (i.e., it is a leaf in the original model). There are $N-1$ scenarios. Let (i, j_1) be the arc under consideration. The solution is: $y^i_{ij_1} = 1$ and $q^t_{ij_1} = 1, \forall t \neq i$. All other variables are zeros. The cost value is $p_{ij_1} + \sum_{t \in V, t \neq i} \mu^t_{[ij_1]}$.

Case 2: Node i has at least two outgoing arcs. Among them, the highest and second highest power levels are of significance. Enumerating over arc pairs representing the two power levels gives $(N-1)(N-2)/2$ scenarios. Denote the two arcs under consideration by (i, j_1) and (i, j_2), respectively, with $p_{ij_1} \geq p_{ij_2}$. The optimum solution is as follows.

- $q^{j_1}_{ij_1} = 1$, $y^{j_1}_{ij_2} = 1$, with cost value $\mu^{j_1}_{[ij_1]} + p_{ij_2}$.
- $q^{j_2}_{ij_2} = 1$, $y^{j_2}_{ij_1} = 1$, with cost value $\mu^{j_2}_{[ij_2]} + p_{ij_1}$.
- $y^i_{ij_1} = 1$, with cost value p_{ij_1}.
- For any $t \neq j_1, j_2, i$, the arc selected to reach t is either (i, j_1), or the arc with the lowest cost value among those having power levels not exceeding p_{ij_2}. The optimum is therefore to select the best of the following two options.
  1. $q^t_{ij_1} = 1$ and $y^t_{ij_2} = 1$, with cost value $\mu^t_{[ij_1]} + p_{ij_2}$.
  2. $q^t_{ik} = 1$ and $y^t_{ij_1} = 1$, with cost value $\mu^t_{[ik]} + p_{ij_1}$, where $k \in \arg\min_{j: p_{ij} \leq p_{ij_2}} \mu^t_{[ij]}$.

Time complexity is an important aspect in problem solution. To solve the Lagrangean sub-problem, the overall time complexity is of $O(N^3)$. This is seen from the fact that $\mu^t_{[ik]}$ depends only on i and j_2 (not on j_1). Hence there are $O(N^2)$ such parameters to compute, and each computation takes $O(N)$ time. All other cost terms are obviously available in $O(N^3)$ time. This is also a lower bound on the time complexity of solving the Lagrangean subproblem, because the solution of each node in S2 is at least quadratic in time complexity. Subproblem S1 is not the computational bottleneck, as computing an MST by Prim's algorithm takes no more than $O(N^2)$ time, and evaluating the broadcast power of all source nodes runs also in $O(N^2)$ time.

For any multiplier vector $\boldsymbol{\mu}$, the Lagrangean function $L_{\mathbf{z}}(\boldsymbol{\mu}) + L_{\mathbf{q},\mathbf{y}}(\boldsymbol{\mu})$ gives a lower estimation of the global optimum. The dual optimum, defined by the multiplier vector μ maximizing the Lagrangean function, is the best possible estimation by dual decomposition. Since the SBT problem is non-convex, there may be a duality gap. Another commonly considered estimation in integer programming is the optimum of the continuous relaxation, obtained by removing the integrality restriction of the variables in an explicit linear formulation. No matter which linear equations and inequalities are used to replace (6), however, the continuous relaxation will not perform better in cost estimation than dual decomposition. The conclusion is formalized below.

Proposition 2 *The continuous relaxation of* (1)–(7) *with* (6) *replaced by any valid linear constraints, does not provide better cost estimation than the optimum of dual decomposition.*

Proof Consider the model where (6) is replaced by some explicit linear constraints. Denote by LP^{EXP} and L^{EXP} the attainable lower bounds obtained by respectively solving the continuous relaxation and applying dual decomposition to the explicit formulation. It follows from the theory of dual decomposition for integer linear optimization (e.g. Theorem 10.3 in [40]), that $L^{\text{EXP}} \geq LP^{\text{EXP}}$. Let L^{IMP} denote the attainable lower bound obtained by the application of dual decomposition to the model with implicit constraints. From the solution procedures for S1 and S2, it is clear that feasible integer vectors are produced, and therefore they satisfy any constraints defining $\mathbf{y} = Q(\mathbf{q})$. Hence, $L^{\text{IMP}} \geq L^{\text{EXP}}$, and the result follows. ∎

For integer linear optimization, the dual function is in general not differentiable. We adopt subgradient optimization to maximize the dual. Let $\bar{\mathbf{z}}$ and $(\bar{\mathbf{q}}, \bar{\mathbf{y}})$ denote the optimum of S1 and S2, respectively. The vector $\bar{\mathbf{g}} = \big(\bar{g}_{ij}^t, (i,j) \in E, t \in V\big)$, where $\bar{g}_{ij}^t = q_{ij}^t + q_{ji}^t - z_{ij}$, defines a subgradient. For updating the multipliers by taking a step in the direction of the subgradient, we apply $\boldsymbol{\mu}' = \boldsymbol{\mu} + \lambda \frac{DualOpt - L_{\mathbf{z}}(\boldsymbol{\mu}) - L_{\mathbf{q},\mathbf{y}}(\boldsymbol{\mu})}{||\bar{\mathbf{g}}||^2} \bar{\mathbf{g}}$, where λ is the step size, which should be between zero and two for asymptotic convergence. Parameter $DualOpt$ denotes the dual optimum. Since $DualOpt$ is not known, it is replaced by the best-known problem solution, that is, the power of the best SBT found so far. Because this value typically exceeds $DualOpt$, it is necessary to gradually reduce the step size λ (e.g., [41]). In Section 6 we detail the rules for step size adjustment in our computational study.

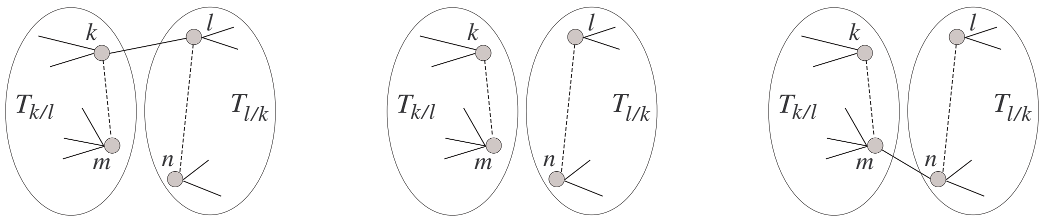
5 Local Search

In subgradient optimization, having a near-optimal SBT value to approximate $DualOpt$ is crucial. Typically, the MSTs originating from S1 are good SBT candidates only when $\boldsymbol{\mu}$ is close to dual optimum. To overcome this issue, we develop a local search algorithm, starting from the MST of subproblem S1. The algorithm adopts the 1-edge exchange operation that deletes a tree edge and adds another edge to merge the two disconnected subtrees into a new spanning tree. The algorithm stops when no 1-edge exchange can lead to any power improvement.

Exchange of one edge is a fundamental algorithmic element for traversing through a sequence of trees in graphs [41]. For MEB, heuristics using a restricted form of 1-edge exchange has been considered in [27]. For SBT, however, this algorithmic operation has not been investigated. From a computational perspective, the major challenge of applying 1-edge exchange to SBT is the complexity of power calculation. For all optimal-tree problems with link-oriented costs, the evaluation of a trial 1-edge exchange runs trivially in $O(1)$ time. Similarly, for MEB and RAP, the new total cost after a 1-edge exchange is available in $O(1)$ time, because only the highest power levels of nodes are of significance, and each of them is counted exactly once. In contrast, in SBT two power levels are significant at all non-leaf nodes, and how many source nodes that respectively use the two levels must be accounted for. In fact, a 1-edge exchange may alter the power expenditure of many nodes. Given an SBT of $N-1$ edges, it is easy to realize that computing the broadcast power for one source (i.e., the cost function of MEB) can be done in $O(N)$ time. Repeating the computation for all sources has $O(N^2)$ time complexity. There can be as many as $O(N^3)$ potential 1-edge exchanges. With the straightforward approach of calculating power from scratch for each source, a single iteration of local search for finding a power-improving tree or concluding its non-existence runs as high as $O(N^5)$ in time.

We present a much faster but non-trivial implementation. To this end, we slightly extend our notation and terminology. Whenever an expression is maximized over some indexed set, we define the maximum value to be zero if the set is empty. When an edge $(k, l) \in T$ is deleted the subtrees $T_{k/l}$ and $T_{l/k}$, where $k \in T_{k/l}$ and $l \in T_{l/k}$, are created. If two distinct nodes are in the same subtree, we say that they are internal to each other, otherwise they are external.

We let $p(i, s)$ denote the power at node i required to forward the message of source s in tree T. If s is internal to i, we define $p'(i, s)$ as the power needed at i to forward the message of s in their joint subtree ($T_{k/l}$ or $T_{l/k}$).



(a) The original tree.

(b) Two subtrees from deleting edge (k, l).

(c) Addition of edge (m, n).

Figure 2: An illustration of one-edge exchange.

Assume that edge (k, l) in T is replaced by a new edge (m, n), where $m \in T_{k/l}$ and $n \in T_{l/k}$, see Figure 2. When computing the new power required for message passing at some node i, exactly three different types of communications need to be considered: message of internal sources forwarded internally, message of external sources forwarded internally, and message of internal sources forwarded both internally and to an external node j. Note that the last type applies only if $i = m$ or $i = n$, and we have $j = n$ if $i = m$ ($j = m$ if $i = n$).

We denote the total power requirements for the three types of message passing by $P^{II}(k, l, m, n)$, $P^{EI}(k, l, m, n)$, and $P^{IE}(k, l, m, n)$, respectively. Note that no node is involved in external forwarding of a message of any external source. This is because there is only one edge, (m,n), joining the two subtrees, and the message does not flow in both directions along the same edge. In the sequel, we explain in detail how to compute $P^{II}(k,l,m,n)$, $P^{EI}(k,l,m,n)$, and $P^{IE}(k,l,m,n)$ for all eligible edges (k,l) and (m,n), such that the time complexity of finding the best 1-edge exchange is kept as low as $O(N^3)$.

5.1 Message of internal source forwarded internally

Message forwarding for an internal source to internal nodes corresponds to viewing each of $T_{l/k}$ and $T_{k/l}$ as an SBT being independent of the other. For internal source s, the forwarding power needed at node i is $p'(i,s)$, and the total power needed for this type of communication becomes

$$P^{II}(k,l,m,n) = \sum_{i \in T_{k/l}} \sum_{s \in T_{k/l}} p'(i,s) + \sum_{i \in T_{l/k}} \sum_{s \in T_{l/k}} p'(i,s). \tag{11}$$

Proposition 3 *Computing $P^{II}(k,l,m,n)$ for all $(k,l) \in T$, $m \in T_{k/l}$ and $n \in T_{l/k}$ has time complexity $O(N^3)$.*

Proof Since $|T| = N-1$ and $P^{II}(k,l,m,n)$ is independent of the new edge (m,n), there are $O(N)$ distinct P^{II}-values to be computed. We show that any $P^{II}(k,l,m,n)$ can be computed in $O(N^2)$ time. Fix $(k,l) \in T$, and choose some $s \in T_{k/l}$. Consider for example a breadth-first traversal of the nodes of $T_{k/l}$ starting at s. When node i is being processed, $p'(i,s)$ is computed as $\max p_{ij}$, where the maximum is taken over all unprocessed neighbors j of i in $T_{k/l}$. It follows that for this choice of s, all $p'(i,s)$ ($i \in T_{k/l}$) are found in $O(N)$ time. Repeating the traversal algorithm for all $s \in T_{k/l}$ shows that the time complexity of computing the first term in (11) is $O(N^2)$. The proof is complete by observing that the same traversal algorithm applied to all $s \in T_{l/k}$ yields the second term in (11). ∎

5.2 Message of external source forwarded internally

Since (n, m) is the only arc from $T_{l/k}$ to $T_{k/l}$, all messages from sources external to $i \in T_{k/l}$ enter $T_{k/l}$ at m. Consequently, we can consider m as a substitute source of the external messages to be broadcasted to nodes in $T_{k/l}$. At node $i \in T_{k/l}$, the power needed for forwarding each message thus becomes $p'(i, m)$. Accounting for all nodes in $T_{k/l}$ and all $\left|T_{l/k}\right|$ external sources, the total power for disseminating broadcast messages in subtree $T_{k/l}$ for external sources becomes $\sum_{i \in T_{k/l}} \left|T_{l/k}\right| p'(i, m)$. Applying the above arguments also to subtree $T_{l/k}$ gives

$$P^{EI}(k, l, m, n) = \sum_{i \in T_{k/l}} \left|T_{l/k}\right| p'(i, m) + \sum_{i \in T_{l/k}} \left|T_{k/l}\right| p'(i, n). \tag{12}$$

Proposition 4 *Computing $P^{EI}(k, l, m, n)$ for all $(k, l) \in T$, $m \in T_{k/l}$ and $n \in T_{l/k}$ has time complexity $O(N^3)$.*

Proof The first term in (12) must be computed for all $(k, l) \in T$ and $m \in T_{k/l}$, but is independent of n. Hence, there are $O(N^2)$ such computations to be made. For fixed $(k, l) \in T$ and $m \in T_{k/l}$, the algorithm suggested in the proof of Proposition 3 can be applied to compute $\sum_{i \in T_{k/l}} \left|T_{l/k}\right| p'(i, m)$ in $O(N)$ time. Consequently, the time complexity of computing the first term of (12) is $O(N^3)$. For the second term, the proof is analogous. ∎

5.3 Message of internal source forwarded internally and externally

In subtree $T_{k/l}$, only node m forwards messages externally to node n in the new tree. The power assigned to m for forwarding message internally may be sufficient to reach node n. If it is not, a power increment is necessary, and in this section we show how this increment is computed.

Assume first that node m has at least two neighbors in $T_{k/l}$. For convenience in notation and without loss of generality, let the end nodes of respectively the most and second most power-requiring edges incident to node m be nodes 1 and 2. If $p_{mn} \leq p_{m2}$, then m has already sufficient power to reach n, no matter from which source in $T_{k/l}$ the message comes. Otherwise, messages that required power p_{m2} for being forwarded internally, will now require an increment to p_{mn} for external forwarding to node n. This applies to all sources that are connected to m via node 1. From the discussion of the definition of SBT in Section 2, the number of message sources requiring the increment can be expressed as $|T_{1/m}|$. If $p_{mn} > p_{m1}$, also messages that needed power p_{m1} for internal forwarding, now call for an increment. The number of sources (see Section 2) from which such messages arrive is $\left|T_{k/l}\right| - |T_{1/m}|$.

Second, if m is a leaf in $T_{k/l}$, the analysis above can be applied by defining $p_{m2} = 0$. Third, if $T_{k/l}$ consists exclusively of node k, implying $m = k$, we define $p_{m1} = p_{m2} = |T_{1/m}| = 0$. Consequently, the power increment required for external message forwarding at node m is

$$P^{IE}_{mn}(k,l) = \begin{cases} 0, & \text{if } p_{mn} \le p_{m2}, \\ |T_{1/m}|\left(p_{mn} - p_{m2}\right), & \text{if } p_{m2} < p_{mn} \le p_{m1}, \\ |T_{1/m}|\left(p_{mn} - p_{m2}\right) + \left(\left|T_{k/l}\right| - |T_{1/m}|\right)\left(p_{mn} - p_{m1}\right), & \text{if } p_{mn} > p_{m1}. \end{cases}$$

By applying the same analysis and parameter definitions to node n, a similar formula for the increment $P^{IE}_{nm}(k,l)$ at n is derived, and we arrive at $P^{IE}(k,l,m,n) = P^{IE}_{mn}(k,l) + P^{IE}_{nm}(k,l)$.

Proposition 5 *Computing $P^{IE}(k,l,m,n)$ for all $(k,l) \in T$, $m \in T_{k/l}$ and $n \in T_{l/k}$ has time complexity $O(N^3)$.*

Proof Since $|T_{1/m}|$, p_{m1}, and p_{m2} depend only on m and (k,l) (not on n), there are $O(N^2)$ such values to be computed. Each of them can be computed in $O(N)$ time. All $P^{IE}_{mn}(k,l)$-values are thus available in $O(N^3)$ time. The proof is complete by a similar observation on $P^{IE}_{nm}(k,l)$. ∎

5.4 The best 1-edge exchange move

We conclude from Sections 5.1-5.3 that when edge (k,l) is replaced by edge (m,n), the new SBT-power is $P^{II}(k,l,m,n) + P^{EI}(k,l,m,n) + P^{IE}(k,l,m,n)$. An iteration of the local search algorithm amounts to enumerating all $(k,l) \in T$ and (m,n) for which $m \in T_{k/l}$ and

$n \in T_{l/k}$, and picking the most power-reducing pair, if any exists. The necessary computations are summarized in an appendix that is available as on-line supplementary material.

Proposition 6 *One iteration of local search for finding the best power-improving 1-edge exchange, or concluding that none exists, runs in* $O(N^3)$ *time.*

Proof Follows directly from Propositions 3-5. ∎

Since there are $O(N^3)$ potential moves to evaluate, the cubic running time is optimal in the sense that no search algorithm based on 1-edge exchange has lower time complexity.

6 Computational Experiments

6.1 Main Results

We report experimental results for 16 network groups. The underlying principle for generating network instances follows that in [1]. The number of nodes varies between 10 and 80. The nodes are randomly placed within a square region (1000×1000 distance units in our case). The power parameter $p_{ij} = d_{ij}^{\alpha}$, where d_{ij} is the Euclidean distance between nodes i and j. The exponent α is set to two and then to four, resulting in two instances per topology. Following the instance-generation specification in [1], there is no restriction on the maximum power, hence the underlying graph is always complete, meaning that for N nodes, the edge set E has size $N(N-1)/2$. A network group is defined by N and α, and each group contains 10 instances.

We first examine the performance of solving the explicit integer programming formulation using (8) by solver CPLEX [42] (version 10.1) on a server with an Opteron processor at 2.4 GHz and 7 GB RAM. For each instance, the time limit is set to 2 hours. Table 1 summarizes the results. For each network group, the table displays the minimum, maximum, and average values of the optimality gap and the computing time. The optimality gap is the relative difference between the power of the best SBT found and the lower cost estimation from the continuous relaxation and the solver's branch-and-bound algorithm. The value is zero if global optimum is reached and proven within the time limit. Sometimes the solver fails to find any integer solution. The gap in this case is denoted by ∞, and excluded from the computation of the average gap value. The computing time is in seconds, and 'lim' is used to denote the time limit. The table contains results for $N \leq 30$. For larger networks, the solution process is time-excessive, and the optimality gap when reaching the time limit has a clearly growing trend in N.

N	$\alpha = 2$		$\alpha = 4$	
	Optimality gap (%) [min, max, ave.]	Time (s) [min, max, ave.]	Optimality gap (%) [min, max, ave.]	Time (s) [min, max, ave.]
10	[0.0, 0.0, 0.0]	[1.0, 10.0, 5.4]	[0.0, 0.0, 0.0]	[1.0, 2.0, 1.3]
20	[6.7, 121.2, 45.8]	[lim, lim, lim]	[0.0, 3.3, 0.3]	[26.0, lim, 1030.4]
30	[84.3, ∞, 223.6]	[lim, lim, lim]	[0.0, ∞, 4205.4]	[lim, lim, lim]

Table 1: Computational results of solving the explicit integer formulation of SBT.

The results in Table 1 show that the explicit integer formulation can be solved rapidly to global optimum only for $N = 10$. For $N = 20$ and $\alpha = 2$, computing global optimum within the time limit becomes out of reach, and the average optimality gap jumps to 45.8%. For this value of α and $N = 30$, there are some test networks for which no integer solution is found at all, and the average gap for the remaining test networks grows substantially to 223.6%. Increasing α to 4 tends to make the computation faster. This is because the power grows much more rapidly with distance in comparison to $\alpha = 2$, thus many of the edges with huge power levels can be easily excluded from further consideration in the solution process. As a result, the average gap is close to zero for $N = 20$ and $\alpha = 4$. When N grows to 30, however, the diversity in the results is very large: For some of the networks, global optimum is attained, whereas for some others no integer solution is found at all, and the average gap is huge. In conclusion, solving the explicit integer formulation does not lead to an efficient approach for getting optimum (except for very small networks), and more importantly, it does not lead to a tight bounding interval.

The dual decomposition algorithm with local search is implemented and run on a laptop (Intel Core i5 with processor speed at 2.53 GHz). In the implementation, subgradient optimization goes through two phases. In the first phase, which is run for K iterations, parameter λ is set to a constant value λ^1. In the second phase, λ is reduced each time by a factor β, if the lower estimation of global optimum does not improve in M consecutive iterations. Each time an improvement in the over estimation (from solving S2 and applying local search) is obtained, λ is set to λ^2. The local search algorithm is embedded into dual decomposition to provide good estimation on $DualOpt$ in subgradient optimization. Local search is applied every L subgradient iterations. All the parameters influence the numerical performance of dual decomposition. For each network group, the parameters can be tuned to produce a group-specific set of values for optimal performance. In order to keep the performance evaluation unbiased, however, we use a fixed set of parameter values for all network groups, with the only exception that parameter K, i.e., the number of iterations of phase one, increases by network size N. For $N \leq 20$, $30 \leq N \leq 40$, $50 \leq N \leq 60$, and $70 \leq N \leq 80$, K is set to $1500, 10000, 25000$, and 35000, respectively. The values of the other parameters are: $\lambda^1 = 2.0$, $\lambda^2 = 1.0$, $\beta = 0.9$, $M = 50$, and $L = 10$. With this parameter setting, the algorithm performs reasonably well for all network groups, although the performance is not the best achievable one for each individual group.

N	$\alpha = 2$		$\alpha = 4$	
	Optimality gap (%) [min, max, ave.]	Time (s) [min, max, ave.]	Optimality gap (%) [min, max, ave.]	Time (s) [min, max, ave.]
10	[0.78, 0.96, 0.90]	[<0.1, 0.2, 0.1]	[0.42, 0.97, 0.77]	[<0.1, 0.2, 0.1]
20	[0.92, 0.99, 0.96]	[0.3, 2.3, 1.2]	[0.73, 1.00, 0.87]	[0.1, 0.3, 0.2]
30	[0.89, 1.75, 1.03]	[8.6, 61.1, 25.0]	[0.65, 1.00, 0.91]	[1.2, 8.0, 2.7]
40	[0.93, 1.94, 1.16]	[22.6, 182.5, 115.7]	[0.87, 1.44, 0.99]	[2.5, 45.4, 11.4]
50	[0.92, 2.70, 1.19]	[93.2, 950.6, 572.1]	[0.82, 1.00, 0.95]	[13.2, 46.5, 27.2]
60	[0.99, 4.50, 2.45]	[1150.4, 2044.5, 1640.2]	[0.90, 1.00, 0.95]	[37.4, 351.0, 109.8]
70	[0.85, 3.29, 1.61]	[2409.3, 4377.8, 3577.1]	[0.87, 1.00, 0.94]	[62.1, 234.4, 144.2]
80	[0.99, 5.85, 2.34]	[4960.5, 7162.3, 6018.3]	[0.92, 1.25, 1.00]	[265.3, 934.3, 570.6]

Table 2: Computational results of the dual decomposition algorithm.

The dual decomposition algorithm uses two stopping criteria. The algorithm terminates if the optimality gap drops below a threshold t, or if $\lambda < \epsilon$ (i.e., no further significant progress is expected because of small step length). In the experiments, $t = 1\%$ and $\epsilon = 0.002$. The optimality gap is defined as the relative difference between the best Lagrangean dual value generated by subgradient optimization, which gives a lower estimation of global optimum, and the power of the best SBT solution found by local search. The results are shown in Table 2.

From Table 2, we observe that the dual decomposition algorithm is very effective. Overall, the performance gap to global optimality is very small, demonstrating that applying dual decomposition to the implicit formulation of SBT produces very tight lower estimation to global optimum, and that the local search algorithm is able to approach close-to-optimal SBT. Consider the results for $\alpha = 2$. For networks with $N \leq 40$, the maximum optimality gap of all networks is below 2%, and the average gap is only 1.2% or less. For $N \geq 50$, the maximum gap becomes larger, but remains below 6%, and the average gap never exceeds 2.45%. The average computing time ranges from a few seconds to a couple of minutes for $N \leq 40$. For $N \geq 50$, the average time grows successively from 10 minutes to about 100 minutes, which remains much more affordable in comparison to bounding the optimum via solving the explicit formulation (cf. Table 1). Note that these results are obtained under the same algorithm parameter setting except for K, and tailoring the parameter setting for each network group is expected to yield further improvement. Finally, it is evident that assuming $\alpha = 4$ in the power formula gives noticeably better results than those for $\alpha = 2$, supporting the aforementioned observation that, empirically, $\alpha = 4$ requires less computational effort. In particular, the maximum and average gap values are constantly below 1.5% and 1%, respectively, and for reaching these values the computing time is more than an order of magnitude shorter than for $\alpha = 2$.

6.2 Additional Experiments

In this section, we report additional experiments for the following analysis. First, we compare the broadcast power consumptions of SBT, MEB, and RAP. Second, the SBT algorithm in [11] is evaluated. Third, for each of the two components in the decomposition algorithm, dual search and local search, we examine its performance without the integration with the other.

From a networking standpoint, a performance comparison between source-independent broadcast by SBT versus that of source-specific MEB trees is of significance. For any individual source node, the power consumption by SBT is greater than or equal to that of the MEB tree. The question is how much higher to expect. A performance assessment via heuristics for SBT and MEB gives good insights but the accuracy is unknown. That dual decomposition provides a very small optimality gap allows for much better accuracy in performance comparison.

A comparison between MEB and SBT is meaningful only if both are used to accomplish the same broadcast communication task. Which strategy is better depends on the type of network and application. If there are very few nodes that will act as potential broadcast sources, source-specific MEB broadcast trees are preferable. When many or all nodes are potential broadcast sources, whether or not creating a large number of MEB trees remains preferable depends on if MEB substantially outperforms SBT in the power consumption. To this end, we consider the scenario in which each node is a broadcast source, and the total power consumption corresponds to that of delivering one broadcast packet originated from every source to the entire network.

The RAP tree targets network connectivity, but it can also be used for source-independent broadcast. In RAP, the performance metric of each node is the power required to reach all its tree neighbors. A naive way of using the RAP tree for broadcast is to let every node use this power to initiate and forward broadcast messages. This approach is however highly inefficient, as it does not utilize the fact that a node does not have to use the maximum power if a packet is transmitted by the corresponding neighbor, and packet forwarding at a leaf is unnecessary unless the packet is initiated by the leaf itself. For the purpose of a fair comparison, we use the RAP tree as an SBT, and evaluate the overall power accordingly.

In Table 3, we provide the comparison between MEB and SBT, and between SBT and RAP. The global optimum of MEB is computed for all source nodes using the formulation in [27]. The global optimal RAP tree is computed using the multi-tree formulation in [43]. The computations are performed for $N \leq 40$, as larger networks require excessive computing time for MEB. The notation "$\frac{\text{SBT}}{\text{MEB}}$" and "$\frac{\text{RAP}}{\text{SBT}}$" denote the performance ratios between the power consumptions of SBT and MEB, and between RAP and SBT, respectively. For SBT, the power is given by the best solution found by dual decomposition; this solution is in average at most 1.2% away from the global optimum for $N \leq 40$, and hence the comparison is highly accurate.

N	[min, max, ave.] of $\frac{\text{SBT}}{\text{MEB}}$		[min, max, ave.] of $\frac{\text{RAP}}{\text{SBT}}$	
	$\alpha = 2$	$\alpha = 4$	$\alpha = 2$	$\alpha = 4$
10	[1.01, 1.17, 1.08]	[1.01, 1.09, 1.05]	[1.00, 1.33, 1.10]	[1.00, 1.13, 1.03]
20	[1.06, 1.16, 1.12]	[1.01, 1.11, 1.05]	[1.06, 1.28, 1.19]	[1.00, 1.24, 1.04]
30	[1.10, 1.21, 1.13]	[1.01, 1.09, 1.04]	[1.06, 1.29, 1.15]	[1.00, 1.12, 1.05]
40	[1.10, 1.20, 1.13]	[1.02, 1.08, 1.04]	[1.08, 1.28, 1.16]	[1.01, 1.12, 1.06]

Table 3: Performance comparison between SBT and MEB, and between SBT and RAP.

From Table 3, it follows that SBT performs well in power in comparison to MEB. For $\alpha = 2$, using one optimal SBT instead of N optimal MEB trees requires at most 21% in extra power consumption. The range of the average values is $[8\%, 13\%]$, amounting to a very moderate power increase. The additional power for $\alpha = 4$ is considerably smaller. This is explained by the fact that, as the link power levels have a very larger diversity, the expensive links are avoided in both SBT and MEB. An interesting observation is that the values do not have any growing trend in the network size N. These observations, along with the SBT's features of having small storage and processing overhead, make SBT a very attractive approach for broadcast.

The RAP tree minimizes the sum of the highest node powers; this optimality criterion, unlike optimal SBT, does not account for which power levels will be used nor how many sources will use each of the power levels, when the tree is used for broadcast. As shown in Table 3, the resulting range of deviation from optimum SBT is, in average, $[10\%, 19\%]$ for $\alpha = 2$. Although the average values are quite moderate, the maximum deviation (33%) is rather significant. As can be expected, the difference between RAP and SBT decreases for $\alpha = 4$. However, even in this case the maximum deviation from optimum remains over 20%.

Along with introducing SBT, the authors of [11] presented an approximation algorithm. We have implemented this SBT algorithm, and made a performance evaluation using the close-to-global-optimal lower bound from dual decomposition. For all the network groups in Table 3, the performance of the algorithm in [11] is very close to that of optimal RAP. Note, however, that the algorithm in [11] is fast, whereas computing optimal RAP is much more demanding. When N grows from 40 to 80, the deviation from optimality increases. More specifically, for

$N = 80$, the average gap reaches 20% and 10%, respectively, for $\alpha = 2$ and $\alpha = 4$. Overall, the algorithm in [11] has reasonable performance in view of its complexity, although it is clearly inferior to the dual decomposition algorithm in terms of optimality gap. We also remark that the authors of [11] have presented a comparative study of the performance of their SBT algorithm and MEB. However, the comparison is more restricted than that in Table 3, because [11] uses the output from the BIP [1] algorithm rather than optimal MEB trees.

The dual decomposition scheme consists in bounding from below (via dualizing constraints) and from above (via subproblem S2 and local search). The two parts are tightly intertwined with each other. From solving the dual subproblem S2, the MST repeatedly forms the starting point of local search. Local search aims at improving $DualOpt$, which is a crucial parameter in optimizing the dual. It is interesting to see the performance of each of the two parts alone, i.e., to examine how the lower-bounding procedure performs without the support of local search, and apply pure local search without having access to the initial solutions from subproblem S2.

N	$\alpha = 2$		$\alpha = 4$	
	Optimality gap (%) [min, max, ave.]	Time (s) [min, max, ave.]	Optimality gap (%) [min, max, ave.]	Time (s) [min, max, ave.]
10	[0.29, 0.99, 0.80]	[<0.1, 0.6, 0.2]	[0.13, 0.97, 0.70]	[<0.1, 0.2, <0.1]
20	[0.05, 0.91, 0.65]	[1.5, 5.6, 3.1]	[0.71, 0.99, 0.90]	[0.1, 0.3, 0.2]
30	[0.28, 1.90, 0.84]	[39.6, 103.0, 59.6]	[0.77, 0.99, 0.92]	[0.83, 15.9, 3.2]
40	[0.20, 1.59, 0.90]	[90.5, 289.3, 191.0]	[0.82, 2.07, 1.03]	[2.0, 47.9, 10.3]
50	[3.75, 15.67, 3.26]	[570.9, 1597.1, 955.1]	[0.77, 0.98, 0.93]	[11.2, 99.8, 27.3]
60	[7.64, 20.04, 7.43]	[1648.1, 3592.0, 2436.1]	[0.72, 3.02, 1.11]	[44.2, 437.8, 135.8]
70	[0.78, 10.59, 4.65]	[3970.7, 6288.0, 5279.8]	[0.82, 0.99, 0.93]	[54.2, 560.0, 253.9]
80	[1.21, 20.96, 9.96]	[7371.6, 10274.6, 8974.8]	[0.86, 1.00, 0.95]	[271.0, 901.3, 612.0]

Table 4: Computational results of pure dual search.

Table 4 shows the results of optimizing the dual without using local search. The algorithm parameter setting is identical to the specification in Section 6.1. It is apparent that the algorithm remains superior in comparison to solving the explicit integer formulation. For network groups with $N \leq 40$ and $\alpha = 2$, as well as for all network groups with $\alpha = 4$, the performance is fully comparable to the original decomposition scheme with local search. In some of these scenarios, the optimality gap is in fact smaller but the corresponding computing time becomes considerably longer. For $\alpha = 2$ and $N \geq 50$, the performance in optimality and computing time are clearly inferior to that of the original decomposition scheme. This is mainly because of the slow convergence caused by poor estimation of $DualOpt$. Thus, embedding local search into subgradient optimization does enhance the convergence in optimizing the dual.

To apply pure local search, we have implemented an SBT-specific greedy construction heuristic for obtaining an initial solution. It starts from a randomly selected node, and augments the tree successively. In each step, the heuristic selects the node and link resulting in the minimum SBT power for all the sources in the partial tree plus the node under examination.

N	[min, max, ave.] for $\alpha = 2$	[min, max, ave.] for $\alpha = 4$
10	[0.00, 2.45, 0.60]	[0.00, 4.83, 0.56]
20	[0.00, 24.77, 17.89]	[0.00, 4.02, 0.95]
30	[5.31, 21.51, 11.62]	[0.00, 6.11, 1.72]
40	[0.00, 111.72, 28.44]	[0.00, 3.28, 0.56]

Table 5: Optimality gap (%) of greedy tree construction and local search.

Table 5 reports the performance of greedy tree construction with local search. The procedure, by itself, does not give any lower estimation of optimum. In the table, the optimality gap is evaluated using the best solution found by the dual decomposition algorithm in its original form. We do not include results for networks with $N > 40$ for the reason that these results do not provide additional insights. Consider the results for $\alpha = 2$. The pure local search algorithm finds close-to-optimal solutions for $N = 10$. For $N \geq 20$, the optimality gap has a clearly growing trend, with high diversity in solution quality. For $N = 40$, the average gap to optimum is almost 30%, and the maximum gap grows above 100%. As expected, for $\alpha = 4$, local search consistently performs close to optimality, yet the maximum gap values are clearly higher than those in Table 2. In conclusion, local search alone does not perform well, and its performance strongly benefits from the starting solutions provided from optimizing the dual.

7 Conclusions

We have presented a computational optimization scheme for minimum-power broadcast in wireless networks using an SBT. The scheme is built upon applying dual decomposition to a discrete optimization formulation that embeds multiple directed trees into an undirected spanning tree. An implicit representation of the connection between some of the variables is preferable to an explicit one, because solving an explicit formulation does not yield an efficient solution approach. By dual decomposition, the implicit formulation is effectively handled. In addition, optimizing the dual is intertwined with a local search algorithm, for which we have presented a time-efficient implementation. Computational experiments demonstrate that our approach enables close-to-optimal solutions. Moreover, the results show that using the optimal SBT instead of many source-specific MEB trees does not lead to significant increase in power expenditure.

That the computational machinery gives very accurate estimation on global optimum is highly useful from a performance assessment standpoint. To this end, one line of further research is the development of distributed algorithms. Another interesting topic consists in problem generalizations, such as SBT in networks using directional antenna, and the use of hop limit to address end-to-end delay.

Acknowledgments

The work of the first author has been supported by the Swedish Research Council, and CENIIT and ELLIIT centers of Linköping University, Sweden. The work of the second author has been supported by the Research Council of Norway under contract no 201434/S10. We thank the anonymous reviewers for their valuable comments and suggestions.

Short Biographies of the Authors

Di Yuan received his PhD degree in optimization at Linköping Institute of Technology, Sweden, in 2001. At present he is full professor in telecommunications at the Department of Science and Technology, Linköping University, and head of a research group in mobile telecommunications. His research interests span design, analysis, and resource optimization of telecommunication systems. In 2011 he works part time at Ericsson Research, Sweden.

Dag Haugland holds a PhD from the Department of Informatics at the University of Bergen, Norway, and is currently full professor in Optimization at the same department. His research covers theoretical work in mathematical programming as well as applications of optimization in the petroleum industry and telecommunications.

References

[1] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proc. IEEE INFOCOM '00*, pages 585–594, 2000.

[2] E. Althaus, G. Călinescu, I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment in ad-hoc wireless networks. *Wireless Networks*, 12:287–299, 2006.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[4] K. Bulbul, O. Ercetin, and T. Ünlüyurt. Hop constrained energy-efficiency broadcasting: insights from massively dense ad hoc networks. *IEEE Transactions on Mobile Computing*, 8:188–202, 2009.

[5] M. Conti and S. Giordano. Multihop ad hoc networks: the reality. *IEEE Communications Magazine*, pages 88–95, April 2007.

[6] C. E. Perkins. *Ad hoc networking*. Addison-Wesley Professional, 2001.

[7] A. K. Das, R. J. Marks, E. El-Sharkawi, P. Arabshahi, and A. Gray. Minimum power broadcast trees for wireless networks: integer programming formulations. In *Proc. IEEE INFOCOM '03*, pages 1001–1110, 2003.

[8] M. Čagalj, J.-P. Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proc. ACM MobiCom '02*, pages 172–182, 2002.

[9] S. Guo and O. Yang. Minimum-energy multicast in wireless ad hoc networks with adaptive antennas: MILP formulations and heuristic algorithms. *IEEE Transactions on Mobile Computing*, 5:333–346, 2006.

[10] P.-J. Wan, G. Călinescu, X.-Y. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *Proc. IEEE INFOCOM '01*, pages 1162–1171, 2001.

[11] I. Papadimitriou and L. Georgiadis. Minimum-energy broadcasting in multi-hop wireless networks using a single broadcast tree. *Mobile Networks and Applications*, 11:361–375, 2006.

[12] R. Montemanni, L. M. Gambardella, and A. K. Das. Models and algorithms for the mpscp: an overview. In J. Wu, editor, *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, pages 133–146. Auerbach Publications, 2006.

[13] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proc. IEEE INFOCOM '00*, pages 404–413, 2000.

[14] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Energy-efficient broadcast and multicast trees in wireless networks. *Mobile Networks and Applications*, 7:481–492, 2002.

[15] C. Ambühl. An optimal bound for the mst algorithm to compute energy efficient broadcast trees in wireless networks. *LNCS*, 3580:1139–1150, 2005.

[16] J. Bauer, D. Haugland, and D. Yuan. New results on the time complexity and approximation ratio of the broadcast incremental power algorithm. *Information Processing Letters*, 109:615–619, 2010.

[17] M. Flammini, R. Klasing, A. Navarra, and S. Perennes. Improved approximation results for the minimum energy broadcasting problem. *Algorithmica*, 49:318–336, 2007.

[18] M. Flammini, R. Klasing, A. Navarra, and S. Perennes. Tightening the upper bound for the minimum energy broadcasting. *Wireless Networks*, 14:659–669, 2008.

[19] R. Klasing, A. Navarra, A. Papadopoulos, and S. Pérennes. Adaptive broadcast consumption (abc), a new heuristic and new bounds for the minimum energy broadcast routing problem. *LNCS*, 2042:866–877, 2004.

[20] A. Navarra. Tighter bounds for the minimum energy broadcasting problem. In *Proc. Third Int'l Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 313–322, 2005.

[21] I. Caragiannis, M. Flammini, C. Kaklamanis, R. Klasing, and A. Navarra. Energy consumption minimization in ad hoc wireless and multi-interface networks. In A. Koster and X. Munoz, editors, *Graphs and Algorithms in Communication Networks*, pages 335–355, Springer-Verlag, 2010.

[22] J. Bauer, D. Haugland, and D. Yuan. A fast local search method for minimum energy broadcast in wireless ad hoc networks. *Operations Research Letters*, 37:75–79, 2009.

[23] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. Power consumption problems in ad-hoc wireless networks. In *Proc. First Workshop on Approximation and Online Algorithms (WAOA)*, pages 252–255, 2004.

[24] I. Kang and R. Poovendran. Iterated local optimization for minimum energy broadcast. In *Proc. Third Int'l Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 332–341, 2005.

[25] W. Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC '02)*, pages 112–122, 2002.

[26] G. D. Nguyen. General algorithms for construction of broadcast and multicast trees with applications to wireless networks. *Journal of Communications and Networks*, 7:263–277, 2005.

[27] D. Yuan, J. Bauer, and D. Haugland. Minimum-energy broadcast and multicast in wireless networks: an integer programming approach and improved heuristic algorithms. *Ad Hoc Networks*, 6:696–717, 2008.

[28] J. Bauer, D. Haugland, and D. Yuan. Analysis and computational study of several integer programming formulations for minimum-energy multicasting in wireless ad hoc networks. *Networks*, 52:57–68, 2008.

[29] M. Min, O. A. Prokopyev, and P. M. Pardalos. Optimal solutions to minimum total energy broadcasting problem in wireless ad hoc networks. *J. Combinatorial Optimization*, 11:59–69, 2006.

[30] A. Clementi, P. Penna, and R. Silvestri. Hardness results for the power range assignment problem in packet radio networks. *LNCS*, 1671:197–208, 1999.

[31] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *LNCS*, 1200:363–374, 1997.

[32] A. Clementi, M. D. Ianni, and R. Silvestri. The minimum broadcast range assignment problem on linear multi-hop wireless networks. *Theoretical Computer Science*, 299:751–761, 2003.

[33] G. K. Das, S. Das, and S. C. Nandy. Range assignment for energy efficient broadcasting in linear radio networks. *Theoretical Computer Science*, 352:332–341, 2006.

[34] R. Montemanni and L. M. Gambardella. Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. *Computers & Operations Research*, 32:2891–2904, 2005.

[35] D. Bein and S. Q. Zheng. Energy efficient all-to-all broadcast in all-wireless networks. *Information Sciences*, 180:1781–1792, 2010.

[36] G. Calinescu, S. Kapoor, and M. Sarwat. Bounded-hops power assignment in ad hoc wireless networks. *Discrete Applied Mathematics*, 154:1358–1371, 2006.

[37] A. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. *Mobile Networks and Applications*, 9:125–140, 2004.

[38] G. K. Das, S. Ghosh, and S. C. Nandy. An efficient heuristic algorithm for 2d h-hops range assignment problem. In *Proc. IEEE GLOBECOM '04*, pages 1051–1055, 2004.

[39] C. Burt, Y.-B Chan, and N. Sonenberg. Exact models for the k-connected minimum energy problem. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 28:392–406, 2010.

[40] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.

[41] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows, Theory, Algorithms, and Applications*. Princeton Hall, 1993.

[42] Cplex 10.1, users manual. ILOG, 2006.

[43] D. Haugland and D. Yuan. Compact integer programming models for optimal tress in wireless networks. In J. Kennington, E. Olinick, and D. Rajan, editors, *Wireless Network Design: Optimization Models and Solution Procedure*, pages 219–246. Springer, 2011.