# A Localized Algorithm Based on Minimum Cost Arborescences for the MECBS Problem with Asymmetric Edge Costs

Frederico Barboza and Flávio Assis

LaSiD - Distributed Systems Laboratory
DCC - Department of Computer Science
UFBA - Federal University of Bahia
Salvador, Bahia, Brazil
fred.barboza@gmail.com, fassis@ufba.br

**Abstract.** In this paper we describe a (distributed) localized approximation algorithm for the MECBS (Minimum Energy Consumption Broadcast Subgraph) problem with asymmetric edge costs, called LMCA (Localized algorithm for energy-efficient broadcast based on **L**ocal **M**inimum **C**ost **A**rborescences). Given a directed weighted graph $G = (V, E)$ with edge weight function $w$ and a source node $s$, the MECBS problem consists of finding a range assignment to $V$ such that the induced graph contains a spanning directed tree rooted at $s$ with minimized cost. This problem can be efficiently solved for some specific cases, but it is NP-hard in the general case. To the best of our knowledge, LMCA is the first localized algorithm to the MECBS problem with asymmetric edge costs (without restricting the way how edge costs might be asymmetric). We compared LMCA with blind flooding and with two alternative solutions we designed for the problem, LMCP and LBIPAsym (a variation of LBIP for the case of asymmetric edge costs). In our experiments, LMCA outperformed these algorithms. We additionally present and evaluate two slight variations of LMCA, called LMCAc and LMCAfl.

**Keywords:** MECBS, asymmetric edge costs, minimum cost arborescence, localized algorithm, wireless sensor networks.

## 1 Introduction

In this paper we address the problem of energy-efficient broadcast in Wireless Sensor Networks (WSN). Broadcast is a useful communication primitive in different scenarios, such as in the dissemination of data or specific requests from the base station to the whole network (for example, for the distribution of cryptographic keys or synchronization packets). Since the need for conserving energy of nodes is a key issue in WSN (due to the fact that the nodes are typically operated by non-replaceable batteries), broadcasting should be performed as efficient in terms of energy use as possible.

D. Simplot-Ryl et al. (Eds.): ADHOCNETS 2011, LNICST 89, pp. 223–238, 2012.

Different approaches for the problem of energy-efficient broadcasting in wireless networks have been proposed. This problem consists of, given a *source node* $s$, determining a transmission power to each node of the network such that: (a) the resulting topology contains a spanning tree rooted at $s$; and (b) the sum of the costs associated with the nodes is minimized. The control of the transmission power of nodes is possible since radio transceivers commonly used in practical systems typically support transmissions at different power levels.

This problem has been formulated as the MECBS (Minimum Energy Consumption Broadcast Subgraph) problem [1]. In this paper we consider a version of this problem in which edge costs might be asymmetric. I.e. transmitting from a node $u$ to a node $v$ might have a different cost than transmitting from $v$ to $u$. This happens, for example, in heterogeneous networks, where transmission and reception costs are dependent on hardware characteristics (specific to each node) or when the cost of *overhearing* is taken into consideration (overhearing is one of the major sources of energy expenditure due to communication in a WSN [14]). Heterogeneous networks have been less considered in the literature on energy-efficient algorithms, but they are the type of networks that is and will be used in many applications. The general MECBS problem is NP-hard [1].

In this paper we describe LMCA (Localized algorithm for energy-efficient broadcast based on **L**ocal **M**inimum **C**ost **A**rborescences), a localized algorithm that is an approximation for the MECBS problem with asymmetric edge costs. Although many results have been proved for symmetric versions of MECBS and some results for the general MECBS problem, to the best of our knowledge LMCA is the first localized algorithm to MECBS with asymmetric edge costs (without restricting the way how edge costs might be asymmetric). As there is no other specific localized algorithm for this problem in the literature, we compared LMCA with blind flooding and with alternative solutions we designed: one based on local computation of trees of minimum cost paths (which we call LMCP) and a variation of LBIP [5] for the case of asymmetric edge costs (which we call LBIPAsym). LBIP was designed for the version of MECBS with symmetric edge costs (for which it is a very good approximation) and thus cannot be directly applied to the version of MECBS we consider in this paper. LMCA outperformed these algorithms in our experiments. We additionally describe small variations of LMCA (which we call, respectively, LMCAc and LMCAfl) which improved slightly on the performance of LMCA in terms of energy cost in some scenarios, but which require, resp., higher processing cost at nodes and the exchange of longer messages.

This paper is structured as follows. In Section 2 we present the MECBS problem. In Section 3 we discuss related work. In Section 4 we describe the adopted system model. In Section 5 we describe LMCA and present proofs of its correctness. In Section 6 we present a performance evaluation of LMCA. Finally, we conclude the paper in Section 7.

## 2 The MECBS Problem

We use the MECBS problem formulation as presented in [1]. Let $G = (V, E)$ be a directed weighted graph with edge weight function $w : E \to \mathbb{R}^+$. A *range assignment* for $G$ is a function $r : V \to \mathbb{R}^+$. The *transmission graph* induced by $G$ and $r$ is defined as $G_r = (V, E')$, where:

$$E' = \cup_{v \in V} \{(v, u) : (v, u) \in E \wedge w(v, u) \leq r(v)\}$$

The MECBS is then defined as follows: given a *source node* $s \in V$, find a range assignment $r$ for $G$ such that $G_r$ contains a directed spanning tree of $G$ rooted at $s$ and $cost(r) = \sum_{v \in V} r(v)$ is minimized.

We consider that the edge weight function might be asymmetric. I.e. for two edges $(u, v)$ and $(v, u)$ in $E$, $w(u, v)$ might be different from $w(v, u)$.

## 3 Related Work

The MECBS problem has been extensively studied. However, most work concentrates on versions of the problem with symmetric edge costs ($w(u, v) = w(v, u)$). Algorithms that are centralized (e.g., BIP [10]), distributed but not localized (e.g., [13]) and localized (RTCP and RBOP [8], LBOP, LBOP-T and RBOP-T [9], TR-LBOP and TRDS [6] and LBIP [5]) for specific versions of the problem have been proposed. In particular, BIP is a very efficient centralized approximation algorithm [5,15] with a $\Omega(n)$ performance ratio [18].

The general MECBS problem is NP-hard [1]. This problem was also proved to be inapproximable within $(1 - \epsilon) \ln n$ for any $\epsilon > 0$, unless $P = NP$ ($n$ denotes the number of nodes) [2,16,1].

For the general case, there are few approximation algorithms. Centralized algorithms were proposed in [18,7,13]. The algorithms presented in [7] and [13] provide logarithmic approximations to MECBS and improves the approximation ratio of the algorithm presented in [18]. The algorithm in [7] provides a $2(ln(n-1)+1)$ approximation ratio. The algorithm in [13] improves slightly on this result, with a $3/2(ln(n-1)+1)$ performance ratio. As there is no sublogarithmic approximation to the problem, these algorithms are asymptotically optimal.

To the best of our knowledge, distributed algorithms that consider asymmetry are only presented in [19,4]. In [19] the authors present a centralized and a distributed algorithm to construct a strongly connected broadcast arborescence with bounded transmission delay. The algorithm is not localized (it is based on distributed algorithms for calculating shortest paths, minimum weight directed spanning trees and depth-first search). In [4], edge costs are defined by multiplying the transmission power by the node's *energy unit cost*, thus restricting the way how edge costs might be different. The algorithm described in the paper (multi-dimensional case, $\alpha > 1$) [4] is not localized as well.

Additionally, in [11], although the authors define a version of the problem with directed graphs and potentially asymmetric edges costs, the distributed algorithm presented is based on the GHS minimum cost spanning tree algorithm,

which assumes undirected graphs. In [15], the authors model the different transmission levels of transceivers (in a homogeneous environment). Asymmetry could be modelled by including all the transmission levels of all radio devices used in a particular setting. However, the number of such levels would be very large if we consider, for example, asymmetry due to hearing costs (the restriction of $\Theta(log(n/logn))$ ranges assumed in the paper would not apply in a general case).

Thus, to the best of our knowledge, LMCA is the first localized approximation algorithm to the MECBS problem with asymmetric edge costs.

## 4 System Model

A wireless sensor network is represented by a strongly connected directed graph $G = (V, E)$ with edge weight function $w : E \to \mathbb{R}^+$, where $V$ is the set of nodes, $E$ represents the set of communication channels and $\mathbb{R}^+$ is the set of nonnegative real numbers. We associate a *process* with each node. A process is a finite state automaton that models the behaviour of a node. Each process has a unique identifier. So we have processes $p_1, p_2, ..., p_n$, where $n = |V|$. One of the processes is the *root* process (or simply *root*). The node associated with the root process is the root node. Since there is a one-to-one relationship between processes and nodes, we will use the terms *process* and *node* interchangeably.

Processes communicate with each other by exchanging messages. Edge $(p, q) \in E$ iff process $q$ can receive messages from process $p$. We assume that each node knows the set of processes in its two-hop neighbourhood. We do not restrict how edge costs are defined. Edge costs might be asymmetric, i.e. $w(p, q)$ might be different from $w(q, p)$, for any two nodes $p$ and $q$. The communication relationship, however, is symmetric (if $u$ can hear a message from $v$, $v$ can also hear a message from $u$, when both transmit at full power). We assume that each node can adjust its transmission power to any value from 0 to a maximum. The graph induced when all nodes transmit at maximum power is strongly connected.

We assume an asynchronous system model. I.e. there is no known upper bound on the time a message takes to arrive at the destination and there is no known upper bound on the time a process takes to execute a single step. Channels and nodes are assumed to be reliable. A message sent by a node eventually arrives without modification at the receiver.

## 5 The LMCA Algorithm

### 5.1 Algorithm Overview

LMCA works incrementally. Starting from the root, each process $p_i$ calculates a minimum cost arborescence rooted at $p_i$ on the graph that represents its two-hop neighbourhood (as explained right below, nodes that are known to have already been covered will be ignored). We use Edmonds's algorithm to find the minimum cost arborescence [3]. Process $p_i$ uses the calculated arborescence to define: (a) the cost associated with it (i.e. $r(p_i)$) and its associated transmission power; (b)

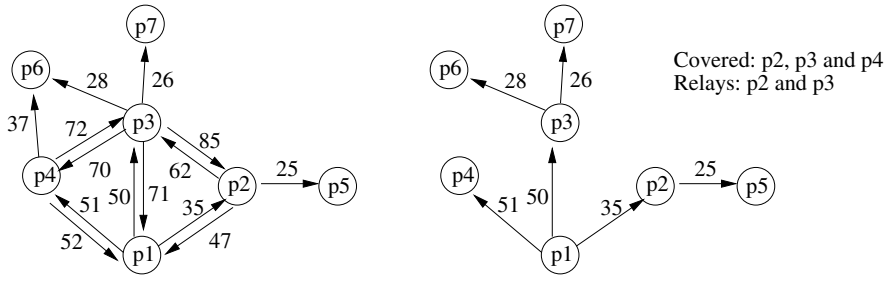



**Fig. 1.** Example of LMCA

the set of processes that become *covered* by it, i.e. that are reached by $p_i$ when it transmits with the defined transmission power; and (c) the next processes to continue the algorithm. These processes are called *relay processes* (or simply *relays*). The algorithm is repeated in the same way by each relay process. The list of processes in $p_i$'s neighbourhood that are known by $p_i$ to have already been covered is transmitted to the next relays. Thus these processes can be ignored in the local calculation of the relays' minimum cost arborescences.

Each node $p_i$ determines the relay and covered processes as follows. Process $p_i$ chooses a process $p_j$ among its children in the minimum cost arborescence for which $w(p_i, p_j)$ is the highest. Let us call this node $highest_i$. Process $p_i$ will adjust its transmission power to the minimum power necessary to reach $highest_i$. All processes $p_k$ in $p_i$'s one-hop neighbourhood for which $w(p_i, p_k)$ is less than or equal to $w(p_i, highest_i)$ are the *processes covered by* $p_i$. The other processes in the arborescence are the *uncovered processes*, according to $p_i$'s view.

We call *bridges* the edges $(q, s)$ in the minimum cost arborescence rooted at $p_i$ which have a covered process as tail $(q)$ and an uncovered process as head $(s)$[1]. The relay processes will be those processes that are tails of bridges.

Figure 1 illustrates how a process $(p_1)$ determines bridges and relays. The left part of the figure shows the graph that represents $p_1$'s two-hop neighbourhood. Processes $p_2$, $p_3$ and $p_4$ are one-hop neighbours of $p_1$. An arrow from node $p_i$ to node $p_j$ represents the edge $(p_i, p_j)$. The numbers beside the arrows represent the costs of the corresponding edges.

The right part of Figure 1 represents the tree built by process $p_1$. Process $p_4$ is $p_1$'s child in the tree for which the cost of the edge from $p_1$ to it is the highest (i.e. $p_4$ is $highest_1$). The transmission power associated with $p_1$ is the minimum power needed to reach $p_4$. The cost associated with $p_1$ is the cost of the $(p_1, p_4)$ edge. The nodes covered by $p_1$ are $p_2$, $p_3$ and $p_4$. Nodes $p_5$, $p_6$ and $p_7$ are uncovered. The bridges are edges $(p_2, p_5)$, $(p_3, p_6)$ and $(p_3, p_7)$. Therefore, the relays will be processes $p_2$ and $p_3$.

After having determined the relays, process $p_i$ broadcasts a RELAY message, using the minimum power needed to reach $highest_i$. This message contains a list of the processes that are known to be covered and those that were chosen as

[1] For an edge $(u, v)$, we call $u$ and $v$, resp., the *tail* and the *head* of the edge.

relays by $p_i$. The nodes that are known by $p_i$ to be covered are those in its two-hop neighbourhood that are covered by $p_i$ or by some other node, as indicated on the list of covered nodes present either in the first RELAY message received by $p_i$ or in an ALREADYRELAY message (see below). Note that all processes that will hear a RELAY message are covered. A process, say $p_j$, that receives this message from process $p_i$ will be in one of the following states:

- $p_j$ *has already been a relay or it is the root*: if $p_i$ chose $p_j$ to be a relay, $p_j$ replies with an ALREADYRELAY message, informing that it has already been a relay. This message contains the set of covered nodes in $p_j$'s two-hop neighbourhood. If $p_i$ did not choose $p_j$ to be a relay, $p_j$ just ignores the message.
- $p_j$ *is not the root, it has not been a relay and it is on the list of relays*: $p_j$ will continue the algorithm (repeating the steps described above with its own local information). $p_i$ becomes the parent of $p_j$.
- $p_j$ *is not the root, it has not been a relay and it is not on the list of relays*: $p_j$ was not chosen by $p_i$ to be a relay. As $p_j$ is covered, the cost associated with it remains zero. $p_i$ becomes the parent of $p_j$.

When process $p_i$ receives an ALREADYRELAY message, it updates the set of covered processes and executes the local procedure again to find a new set of covered nodes, relays and a new transmission power. The new transmission power will be the maximum between the current and the new one.

This algorithm is presented in Figure 2, where $p_i$ represents a generic process. All processes execute the same algorithm. We assume that $p_i$ already knows the $G_i^{2h} = (V_i^{2h}, E_i^{2h})$ graph, i.e. the graph that represents its two-hop neighbourhood. Procedure FINDLOCALMCA represents the main part of the algorithm: elimination of nodes already covered (Fig. 2, lines 1-2); calculation of the local minimum cost arborescence rooted at $p_i$ (represented by the EDMONDS procedure in Fig. 2, line 3); determination of $highest_i$ (Fig. 2, line 4); determination of $p_i$'s cost (Fig. 2, line 5); determination of the new covered and uncovered nodes (Fig. 2, lines 6-7), bridges and relays (Fig. 2, lines 8-9); update of the set of covered nodes in $p_i$'s two-hop neighbourhood (Fig. 2, line 10); and the broadcast of a RELAY message (Fig. 2, line 11), using the power associated with the current value of $myCost_i$.

Lines 12-14 of Fig. 2 represent the actions performed by $p_i$ on starting the algorithm. Lines 15-17 of Fig. 2 represent the actions performed by $p_i$ when it receives an ALREADYRELAY message. Lines 18-25 of Fig. 2 represent the actions performed by $p_i$ when it receives a RELAY message. It either: replies with an ALREADYRELAY message; executes FINDLOCALMCA and sets its parent; or simply updates its parent, as described above.

**Full Coverage**

The algorithm, as described above, does not guarantee full coverage (i.e. that all nodes become covered at the end of the algorithm). In our experiments (see Section 6), however, the algorithm did not cover all nodes only in very rare

Let $G_i^{2h} = (V_i^{2h}, E_i^{2h})$ be the graph that represents $p_i$'s two-hop neighbourhood
$OneHopNeighbours_i \leftarrow \{q : (q \in V_i^{2h}) \wedge ((p_i, q) \in E_i^{2h})\}$
$Covered_i \leftarrow \emptyset \quad myCost_i \leftarrow 0 \quad parent_i \leftarrow nil \quad NewCovered_i \leftarrow \emptyset$
$IAmRelay_i \leftarrow false$

**Procedure** FINDLOCALMCA
1 $V_i \leftarrow (V_i^{2h} \setminus Covered_i) \cup \{p_i\}$
2 $E_i \leftarrow E_i^{2h} \setminus \{(q, r) : ((q, r) \in E_i^{2h}) \wedge ((q \notin V_i) \vee (r \notin V_i))\}$
3 $MCA_i(V_{MCA_i}, E_{MCA_i}) \leftarrow$ EDMONDS $(p_i, G_i)$
4 $highest_i \leftarrow$ any member of the set $\{q : ((p_i, q) \in E_{MCA_i}) \wedge (\nexists r : ((p_i, r) \in E_{MCA_i}) \wedge (cost(p_i, r) > cost(p_i, q)))\}$
5 $myCost_i \leftarrow max\{myCost_i, cost(p_i, highest_i)\}$
6 $NewCovered_i \leftarrow \{q : (q \in OneHopNeighbours_i) \wedge (cost(p_i, q) \leq myCost_i)\}$
7 $Uncovered_i \leftarrow V_i \setminus (NewCovered_i \cup \{p_i\})$
8 $Bridges_i \leftarrow \{(q, s) : ((q, s) \in E_{MCA_i}) \wedge (q \in NewCovered_i) \wedge (s \in Uncovered_i)\}$
9 $RelayProcs_i \leftarrow \{q : (q \in NewCovered_i) \wedge (\exists r \in V_i : (q, r) \in Bridges_i)\}$
10 $Covered_i \leftarrow (NewCovered_i \cup Covered_i) \cap V_i^{2h}$
11 Send RELAY($Covered_i$, $RelayProcs_i$) using power defined by $myCost_i$
**end**

12 **on** starting LMCA
13 **if** $((p_i = root) \wedge (\exists$ *uncovered node in* $OneHopNeighbours_i))$ **then**
14 FINDLOCALMCA
**end**

15 **on** receiving ALREADYRELAY ($pCvrd$) from $p_j$
16 $Covered_i \leftarrow Covered_i \cup pCvrd$
17 **if** ($\exists$ *uncovered node in* $OneHopNeighbours_i$) **then** FINDLOCALMCA
**end**

18 **on** receiving RELAY ($pCvrd$, $pRelays$) from $p_j$
19 **if** $((p_i = root) \vee (IAmRelay_i))$ **then**
20 **if** $(p_i \in pRelays)$ **then** Send ALREADYRELAY($Covered_i$) to $p_j$
**else**
21 **if** $(p_i \in pRelays)$ **then**
22 $Covered_i \leftarrow pCvrd$
23 $IAmRelay_i \leftarrow true$
24 FINDLOCALMCA
25 $parent_i \leftarrow p_j$
**end**

**Fig. 2.** Algorithm executed by process $p_i$ (Part I) - Main part

situations. Full coverage can be guaranteed by making each node check (after a certain period of time) if it has been covered. If it has not, it simply asks one of the covered nodes in its one-hop neighbourhood for covering it. The chosen neighbour adjusts its transmission power in order to cover the requesting

$OneHopNeighbours_i$ and $myCost_i$ are as defined in Fig. 2.

```
1  on timeout { Only executed if p_i is not covered yet }
2      CoveredNeighs_i ← {q : (q ∈ OneHopNeighbours_i) ∧ (q is covered)}
3      if (CoveredNeighs_i = ∅) then
4          restart timeout
       else
5          RelayNeighs_i ← {q : (q ∈ CoveredNeighs_i) ∧ (q is a relay)}
6          if (RelayNeighs_i ≠ ∅) then
7              p_r ← any member of RelayNeighs_i
           else
8              p_r ← any member of CoveredNeighs_i
9          send CoverRequest to p_r
10         parent_i ← p_r
   end
11 on receiving CoverRequest from p_j
12     myCost_i ← max{myCost_i, cost(p_i, p_j)}
13     Covered_i ← {q : (q ∈ OneHopNeighbours_i) ∧ (cost(p_i, q) ≤ myCost_i)}
14     IAmRelay_i ← true        { p_i becomes a relay (if it has not been yet) }
   end
```

**Fig. 3.** Algorithm executed by process $p_i$ (Part II) - Enforcing coverage

node and thus becomes a relay (if it has not been one yet). It is shown in Section 5.2 that any uncovered node will eventually have a covered node in its neighbourhood.

This extension is represented in Figure 3, for a generic process $p_i$. While a process is not covered yet, it periodically verifies if some of its neighbours has already been covered (Fig. 3, lines 1-4). The set of covered neighbours is represented by $CoveredNeighs_i$ (Fig. 3, line 2). If $CoveredNeighs_i$ is empty, the process simply restarts a timer, to execute this verification again in the future (Fig. 3, lines 3-4). Otherwise, process $p_i$ chooses a covered neighbour to cover it. It chooses one neighbour that is a relay, if there is any (Fig. 3, lines 5-7). The set of relay nodes is represented by $RelayNeighs_i$. If there is not any, it chooses one of the covered (but not relay) nodes (Fig. 3, line 8). The chosen neighbour, represented by $p_r$, should be one that minimizes the overall broadcast cost. Process $p_i$ sends a message to process $p_r$, asking $p_r$ for adjusting its power to cover it (Fig. 3, line 9). Process $p_r$ becomes $p_i$'s parent (Fig. 3, line 10).

When a process $p_i$ receives a CoverRequest message from a process $p_j$, it adjusts its power to cover $p_j$ (Fig. 3, lines 11-12). Process $p_i$ updates its set of covered nodes and becomes a relay, if it is not one yet (Fig. 3, lines 13-14).

A process $p_i$ obtains information about which of its neighbours have been covered or are relays by exchanging specific messages with its neighbours. We do not specify this message exchange here.

**Induced Graph**

LMCA induces a graph, $G_r = (V, E_r)$, where:

$$E_r = \bigcup_{p \in V} \{(p,q) : ((p,q) \in E) \wedge (w(p,q) \leq myCost_p)\}$$

I.e. $G_r$ contains all processes as vertices and all edges $(p, q)$ such that node $p$ covers node $q$. This graph might be different from a minimum cost arborescence of the whole original graph. It might contain more than one edge with the same node as head.

Observe that LMCA uses only knowledge about its two-hop neighbourhood, so it is localized. Note that a message to be broadcast by an application can be piggybacked in the RELAY messages or forwarded to a node as a reply to a COVERREQUEST message (thus avoiding an extra delay and overhead to execute LMCA before actually broadcasting an application message).

The cost of the algorithm executed locally by each process is dominated by EDMONDS. Edmonds's algorithm runs in $O(E_i^{2h} + V_i^{2h} log V_i^{2h})$ time [3].

### 5.2 Correctness

LMCA shall satisfy termination, full coverage and the induced graph, $G_r$, must contain a directed spanning tree rooted at the root node. These properties are represented by the lemmas and theorems presented in this section.

We refer to the two parts of LMCA as: (a) Part I, represented in Figure 2, where processes incrementally calculate local minimum cost arborescences and determine covered and relay nodes; and (b) Part II, represented in Figure 3, which is needed to guarantee full coverage of nodes.

**Lemma 1.** *Eventually all processes become covered.*

*Proof.* Full coverage is simply guaranteed by Part II of LMCA. Suppose, by contradiction, that the execution of the algorithm reaches a (global) final state where some of the nodes are covered and some are not (we assume, for simplicity, that the root node is covered at the beginning of the algorithm - so there is always at least one covered node). As the graph induced when all nodes transmit at maximum power is strongly connected, there would be at least one uncovered node that is neighbour of a covered one in this graph. As a node executes Part II periodically until it becomes covered, all uncovered nodes that are neighbours of covered nodes will become covered. By repeating this argument, all nodes will eventually become covered.

**Lemma 2.** *LMCA eventually terminates, i.e. each process reaches a final state, at which no transitions are possible.*

*Proof.* As all nodes eventually become covered (lemma 1), in all executions of LMCA each node is either (a) relay (including the root) or (b) a covered but non-relay node. In case (a), the execution of the node terminates, because each node executes FINDLOCALMCA only a limited number of times as,

each time a process (re-)executes it, it either chooses new relays or eliminates at least one of its neighbours (which was chosen relay, but has already been one). By this argument, each node might only receive a bounded number of RELAY messages. Furthermore, each node might receive a bounded number of COVERREQUEST messages as well (each one-hop neighbour that sends such a message does it only once, as channels and processes are reliable). In case (b), a covered non-relay node becomes covered either passively (as a consequence of the execution of FINDLOCALMCA by some relay node) or actively (by executing Part II of LMCA). Thus, each node receives at most a bounded number of messages (RELAY messages from its relay neighbours).

**Theorem 1.** *When LMCA terminates, the graph induced by it, $G_r = (V, E_r)$, contains an arborescence rooted at the root node.*

*Proof.* We must show that there is a path in $G_r$ from the root node to each other node in the graph. As all nodes are covered, all nodes will have a parent node, except the root. First we show that $G_r$ does not have any cycle. Suppose, by contradiction, that there is a cycle $\langle p_{i_1}, p_{i_2}, ..., p_{i_m} \rangle$, where $p_{i_1} = p_{i_m}$, $m \geq 2$, and $p_{i_j}$ is the parent of $p_{i_{j+1}}$. As all nodes in the cycle are parents of some other node, they are all relays. As $p_{i_k}$ is parent of $p_{i_{k+1}}$, $p_{i_k}$ became a relay before $p_{i_{k+1}}$. Thus, transitively, $p_{i_1}$ became relay before $p_{i_m}$ which became relay before $p_{i_1}$ (a contradiction). As there are no cycles in $G_r$, the reverse paths $R = \langle q_{i_1}, q_{i_2}, ..., q_{i_r} \rangle$, where $q_{i_k+1}$ is the parent of $q_{i_k}$ in $G_r$ ($k \geq 1$) ends at the root node (i.e. $q_{i_r}$ is the root node, since it is the only node that has no parent). Thus, there is a path from the root node to each node in the graph (just follow $R$ in the opposite direction).

## 6 Evaluation

### 6.1 Compared Algorithms

As discussed in Section 3, we are not aware of any other localized algorithm for the problem of MECBS with asymmetric edge costs, as considered in this paper. Therefore in order to evaluate the performance of LMCA we compared it with blind flooding and with two alternative solutions we designed, as variations of, respectively, LMCA and LBIP [5]. We call them LMCP (Localized algorithm for energy-efficient broadcast based on **L**ocal **M**inimum-**C**ost **P**aths trees) and LBIPAsym (**LBIP** for graphs with **Asym**metric edge costs).

In blind flooding each node that receives a message simply broadcasts it with maximum power. It corresponds to a situation with no specific power control.

LMCP is a simple variation of LMCA where each node $p_i$ calculates a tree of minimum cost paths (from itself to nodes in its two-hop neighbourhood) instead of a minimum cost arborescence. It can be proved that LMCP guarantees full coverage of the graph. Let $T_{sp} = (V, E_{sp})$ be a shortest-paths tree rooted at the root node, calculated over a maximum power graph. The proof is based on the fact that the edges in $T_{sp}$ will belong to the trees calculated locally by the nodes.

LBIPAsym is a variation of LBIP [5]. LBIP is known as a very good approximation algorithm for the version of MECBS with symmetric edge costs. As we are assuming that edge costs might be asymmetric, LBIP cannot be directly applied. In LBIP, each process constructs locally a tree (as in LMCA) and each tree induces a broadcast cost. This local tree is constructed iteratively as follows. For a set of already covered nodes, a new uncovered node $u$ is inserted in the tree that is: adjacent to a covered node, say $v$; and the edge connecting $u$ and $v$ is the one which results in the lowest increment in the broadcast tree so far. In LBIPAsym, the only difference is that we consider only the directed edges from covered nodes to uncovered nodes when choosing a new node to be inserted in the tree.

We additionally investigated the impact of modifying LMCA in two different ways, generating two slight variations of the algorithm, which we call, resp., LMCAc and LMCAfl. LMCAc performs an additional processing when determining the relay nodes, in order to try to decrease their number. In LMCAfl, when a node $p_i$ sends a RELAY message, this message will now contain a list with *all* nodes that $p_i$ knows to have been covered so far, i.e. the list might now include nodes that are not in $p_i$'s two-hop neighbourhood.

### 6.2 Description of the Experiments

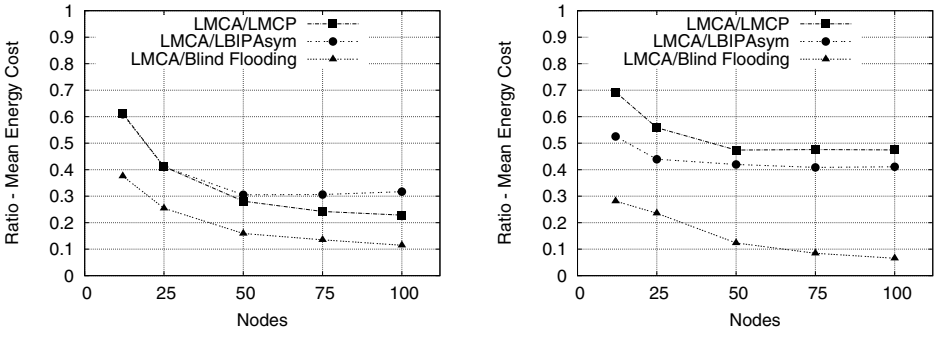
As in this paper we are interested in the cost of the final range assignment, we have implemented a specific Java program for the experiments. The nodes were distributed over a 500m x 500m area. We varied the density of the network. We made experiments with 12, 25, 50, 75 and 100 nodes, randomly spread over the area. All nodes are stationary. For each network configuration (number of nodes), 100 different scenarios were generated.

LMCA is independent of any specific edge cost function. For the experiments, we adopted a cost function based on the energy model described in [17]. Energy is spent by nodes during transmission and reception states (the energy spent during processing is ignored).

For each $(u, v)$, $cost(u, v)$ denotes the energy spent by the network when $u$ transmits with the minimum power necessary to reach node $v$. We consider that this cost involves the energy spent by $u$ to transmit and the energy spent by all nodes that hear the transmission (reception cost). $cost(u, v)$ is thus defined as:

$$cost(u,v) = cf(u) + \ \gamma(u).d(u,v)^{\alpha} + \sum_{\forall s:(s\neq u)\wedge(d(u,s)\leq d(u,v))} cr(s) \qquad (1)$$

where: $cf(u)$ is the (fixed) energy spent by the transmitter electronics at node $u$; $\gamma(u)$ is a parameter characteristic of the transceiver and the channel [12]; $d(u, v)$ is the distance (in meter) between $u$ and $v$; $\alpha$ is the path loss exponent $(2 \leq \alpha \leq 6)$ [12]; and $cr(s)$ is the reception cost of node $s$.



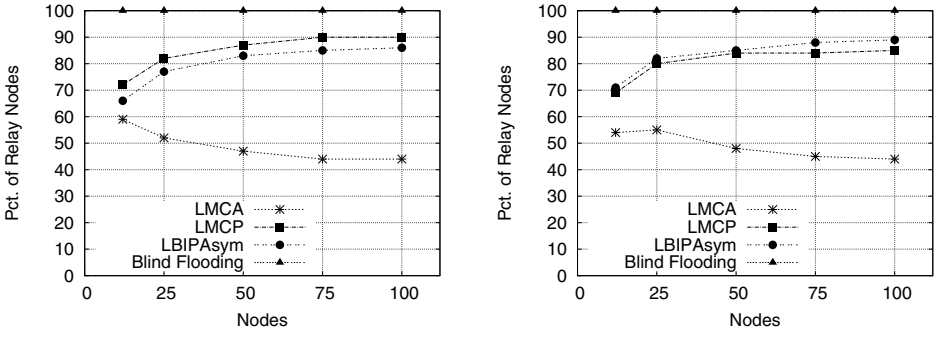

(a) $\alpha = 2$ (b) $\alpha = 6$

**Fig. 4.** Ratio of the mean energy cost of LMCA and, resp., LMCP, LBIPAsym and Blind Flooding

We used the following values: $cf(u) = 48$ nJ/bit, $\gamma(u) = 16.1$ pJ/bit/$m^2$ and $cr(v) = 236.4$ nJ/bit. For each scenario, we used first $\alpha = 2$ and then $\alpha = 6$. Maximum transmission range was 30 units. These values were based on characteristics of the CC2420 transceiver. Since the reception cost of nodes is used in the cost function, $cost(u, v)$ and $cost(v, u)$ might be different (asymmetric costs).

### 6.3 Experiment Results

First we describe a comparison between LMCA and blind flooding, LMCP and LBIPAsym in relation to energy cost (see Section 2). Figure 4 shows the ratio between the mean energy cost of LMCA and, respectively, the mean energy cost of LMCP (line with squares), LBIPAsym (line with circles) and blind flooding (line with triangles). Figure 4(a) represents the results of the experiments for $\alpha = 2$. Figure 4(b) represents the results of the experiments for $\alpha = 6$. Each point in the graphic represents the ratio between the mean energy cost of LMCA and each of the other algorithms, considering 100 scenarios. We see that LMCA outperformed LMCP, LBIPAsym and blind flooding. For $\alpha = 2$, the ratio decreases with the increase in network density, with the exception of a very little increase in the ratio between LMCA and LBIPAsym from 50 to 100 nodes (from 0.30 to 0.31). For example, in the scenarios with 100 nodes, the cost of LMCA was 11.5% of the cost of blind flooding, 22.8% of the cost of LMCP and 32.0% of the cost of LBIPAsym. For $\alpha = 6$, the ratio decreased with the increase in network density, but the ratio between LMCA and, resp., LMCP and LBIPAsym remained stable for 50 or more nodes. Thus, according to the experiments, calculating locally a minimum cost arborescence (LMCA) provided better results than calculating locally trees of minimum cost paths (LMCP). The adaptation of LBIP to graph with asymmetric edge costs performed worse than LMCA as well.




(a) $\alpha = 2$ (b) $\alpha = 6$

**Fig. 5.** Mean percentage of relays: LMCA, LMCP, LBIPAsym and Blind Flooding

Figure 5 represents the percentage of relays for LMCA, LMCP, LBIPAsym and blind flooding (for $\alpha = 2$ and $\alpha = 6$). The percentage of relays for blind flooding is always 100%, as each process locally broadcasts a received message. We see that, differently from LMCP and LBIPAsym, in LMCA the percentage of relay nodes decreases with the increase in the number of nodes in the network. The behaviour of the algorithms for the cases of $\alpha = 2$ and $\alpha = 6$ was similar.

We additionally evaluated whether there would be an improvement in the results of LMCA if (a) we tried to decrease the number of relays by considering the coverage of nodes which are relay candidates and (b) by passing more information about covered nodes between relays. As previously stated, we call these variations LMCAc and LMCAfl, respectively.

More specifically, the difference between LMCAc and LMCA is the following. Recall that $RelayProcs_i$ is the set of relays chosen by process $p_i$ in LMCA (see Figure 2). In LMCAc, process $p_i$ further processes $RelayProcs_i$ removing from this set nodes whose children in the local minimum cost arborescence are covered by other nodes in the set. I.e. a node $v$ is removed from $RelayProcs_i$ if there is another node $u$ in the set that covers $v$'s children when $u$ transmits with the power defined by LMCA (induced by the local tree). This process is illustrated in Figure 6. This figure represents the tree locally built by node $s$ (minimum cost arborescence rooted at $s$). According to LMCA, processes $u$ and $v$ will be relays. But, as the range of $u$ includes $v$'s children, $v$ is removed from the set of chosen relays ($RelayProcs_i$).

In LMCAfl, each node $p_i$, instead of informing to the next relays the list of covered nodes in $p_i$'s two-hop neighbourhood, it includes in this list the set of *all nodes* that $p_i$ knows to have already been covered (so far, in the execution of the algorithm).

Figure 7 shows the ratios of the mean energy cost, number of exchanged messages, size of the list of covered nodes and percentage of relay nodes between LMCA and LMCAc, for the cases of $\alpha = 2$ and $\alpha = 6$. For $\alpha = 2$, all the ratios

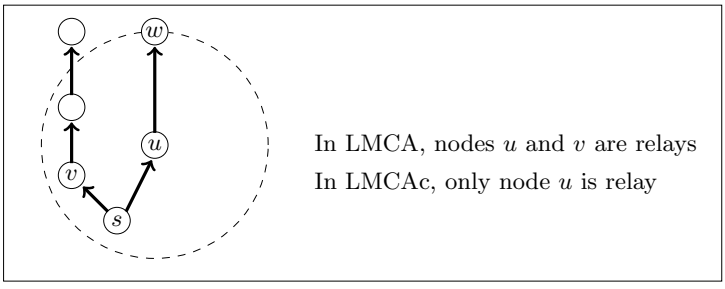
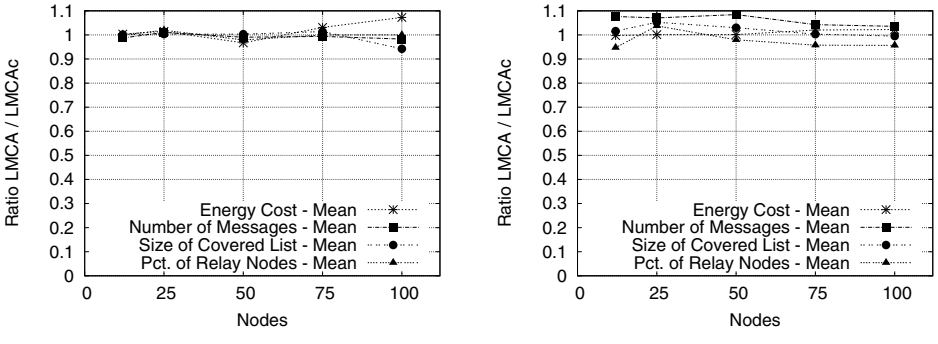



**Fig. 6.** Example of LMCAc (tree built at node $s$)

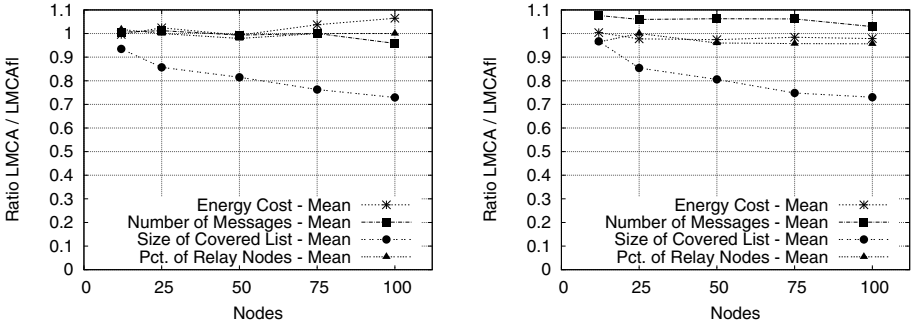



(a) $\alpha = 2$ (b) $\alpha = 6$

**Fig. 7.** Impact of further processing the set of candidate relay nodes (LMCAc)

are very close to 1, except for the case of 100 nodes, where the mean size of the list of covered nodes is smaller for LMCA and LMCAc exhibited smaller energy cost. For $\alpha = 6$, the ratios of the mean number of messages, size of the list of covered nodes and percentage of relay nodes varied slightly more than for the case of $\alpha = 2$. These ratios, however, remained in the range between 0.95 and 1.1. The ratios related to energy cost, however, remained very close to 1. Thus, the performance in terms of energy cost was practically the same for both algorithms, but LMCA exchanged more messages. Recall that LMCAc requires more local processing at each node.

Figure 8 shows the ratios of the same parameters (total energy cost, number of exchanged messages, size of the list of covered nodes and percentage of relay nodes), but now for LMCA and LMCAfl. For both values of $\alpha$ (2 and 6), the mean size of the list of covered nodes was higher for LMCAfl (as expected) and the mean percentages of relay nodes used by LMCA and LMCAfl were very close. For $\alpha = 2$, LMCAfl resulted in a mean energy cost that is either very close or lower than the mean energy cost of LMCA. LMCA, however, had a light improvement on the number of exchanged messages, when the number of nodes was 100. For $\alpha = 6$, the energy cost was roughly the same, but LMCA




(a) $\alpha = 2$ (b) $\alpha = 6$

**Fig. 8.** Impact of transferring more information about covered nodes between relays (LMCAfl)

resulted in more message exchange (although the ratio was always less than 1.1). The performances of the algorithms were equivalent, but LMCAfl requires longer messages to be sent between nodes.

## 7 Conclusion

In this paper, we presented LMCA, a localized algorithm for the MECBS problem with asymmetric edge costs. We are not aware of any other localized algorithm for this version of the problem. We think that this version of MECBS is very relevant, as it models scenarios in heterogeneous sensor networks.

To evaluate the performance of LMCA, we compared it with blind flooding and variations of LMCA and LBIP that we designed, called resp. LMCP and LBIPAsym. In our experiments, LMCA outperformed these algorithms, achieving 10% and 8%, 22% and 48%, and 32% and 41% of the costs of, respectively, blind flooding, LMCP and LBIPAsym, for the cases of $\alpha = 2$ and $\alpha = 6$.

We analysed additionally the impact of further refining the choice of relay nodes, by taking into consideration the coverage of relay candidates, and of transferring more information about covered nodes between relays. The performance of these variations was slight better in some scenarios, but at the cost of either further processing by the nodes or the exchange of longer messages. For $\alpha = 6$, the performances of these algorithms and LMCA were equivalent.

## References

1. Clementi, A.E.F., Crescenzi, P., Penna, P., Rossi, G., Vocca, P.: On the Complexity of Computing Minimum Energy Consumption Broadcast Subgraphs. In: Ferreira, A., Reichel, H. (eds.) STACS 2001. LNCS, vol. 2010, pp. 121–131. Springer, Heidelberg (2001)

2. Clementi, A.E.F., Crescenzi, P., Penna, P., Rossi, G., Vocca, P.: A worst-case analysis of an MST-based heuristic to construct energy-efficient broadcast trees in wireless networks. Technical Report 010, University of Rome "Tor Vergata", Math Department (2001)
3. Korte, B., Vygen, J.: Combinatorial Optimization - Theory and Algorithms, 4th edn. Springer (2008)
4. Ambühl, C., Clementi, A.E.F., Ianni, M., Rossi, G., Monti, A., Silvestri, R.: The range assignment problem in non-homogeneous static ad-hoc networks. In: Proc. of IPDPS 2004 (2004)
5. Ingelrest, F., Simplot-Ryl, D.: Localized broadcast incremental power protocol for wireless ad hoc networks. Wireless Networks 14 (2008)
6. Ingelrest, F., Simplot-Ryl, D., Stojmenović, I.: Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks. IEEE Trans. on Parallel and Distr. Systems 17(6) (June 2006)
7. Calinescu, G., Kapoor, S., Olshevsky, A., Zelikovsky, A.: Network Lifetime and Power Assignment in ad hoc Wireless Networks. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 114–126. Springer, Heidelberg (2003)
8. Cartigny, J., Simplot-Ryl, D., Stojmenovic, I.: Localized minimum-energy broadcasting in ad-hoc networks. In: Procs. of INFOCOM (2003)
9. Cartigny, J., Ingelrest, F., Simplot-Ryl, D., Stojmenovic, I.: Localized LMST and RNG based minimum-energy broadcast protocols in ad hoc networks. Ad Hoc Networks 3 (2005)
10. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: Energy-efficient broadcast and multicast trees in wireless networks. Mobile Network and Applications 7(6), 481–492 (2002)
11. Čagalj, M., Hubaux, J.-P., Enz, C.: Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues. In: MobiCom 2002, pp. 172–182. ACM, New York (2002)
12. Rappaport, T.S.: Wireless Communications: Principles and Practice, 2nd edn. Prentice-Hall (1996)
13. Ghosh, S.K.: Energy efficient broadcast in distributed ad-hoc wireless networks. In: Procs of the 11th IEEE Int. Conf. on Computational Science and Engineering (CSE 2008), São Paulo, Brazil (July 2008)
14. Misra, S., Mohanta, D.: Adaptive listen for energy-efficient medium access control in wireless sensor networks. Multimedia Tools and Applications 47(1), 121–145 (2010)
15. Calamoneri, T., Clementi, A., Monti, A., Rossi, G., Silvestri, R.: Minimum-energy broadcast in random-grid ad-hoc networks: Approximation and distributed algorithms. In: Procs. of MSWiM 2008. ACM Press (2008)
16. Wan, P.-J., Calinescu, G., Li, X.-Y., Frieder, O.: Minimum-energy broadcasting in static ad hoc wireless networks. Wireless Networks 8, 607–617 (2002)
17. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. on Wireless Comms 1(4), 660–670 (2002)
18. Liang, W.: Constructing minimum-energy broadcast trees in wireless ad hoc networks. In: Procs. of MOBIHOC 2002, Switzerland. ACM (June 2002)
19. Li, Y., Thai, M.T., Wang, F., Du, D.-Z.: On the construction of a strongly connected broadcast arborescence with bounded transmission delay. IEEE Transactions on Mobile Computing 5(10), 1460–1470 (2006)