New results on the time complexity and approximation ratio of the Broadcast Incremental Power algorithm

Joanna Bauer, Dag Haugland and Di Yuan

N.B.: When citing this work, cite the original article.

Original Publication:
http://dx.doi.org/10.1016/j.ipl.2009.02.012
Copyright: Elsevier Science B.V., Amsterdam.
http://www.elsevier.com/

Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-18149
New Results on the Time Complexity and Approximation Ratio of the Broadcast Incremental Power Algorithm

Joanna Bauer\textsuperscript{a,1}, Dag Haugland\textsuperscript{a,1,*}, Di Yuan\textsuperscript{b,2}

\textsuperscript{a} Department of Informatics, University of Bergen, PB. 7800, 5020 Bergen, Norway, Tel: +47 55584033, Fax: +47 55584199
\textsuperscript{b} Department of Science and Technology, Linköping University, 60174 Norrköping, Sweden

Abstract

The Broadcast Incremental Power (BIP) algorithm is the most frequently cited method for the minimum energy broadcast routing problem. A recent survey concluded that BIP has $O(|V|^3)$ time complexity, and that its approximation ratio is at least 4.33. We strengthen these results to $O(|V|^2)$ and 4.598, respectively.

Key words: Approximation Algorithms, Analysis of Algorithms, Wireless Ad hoc Network, Minimum Energy Broadcast.

* Corresponding author
Email addresses: Joanna.Bauer@ii.uib.no (Joanna Bauer), Dag.Haugland@ii.uib.no (Dag Haugland), diyua@itn.liu.se (Di Yuan).
\textsuperscript{1} Supported by the Research Council of Norway under contract 160233/V30
\textsuperscript{2} Supported by the Swedish Research Council under grant 621-2004-3902

Preprint submitted to Information Processing Letters 14 May 2009
1 Introduction

In many applications of wireless ad hoc systems, a minimum energy broadcast routing has to be computed repeatedly and quickly. To establish a broadcast routing, a transmission power must be assigned to each network unit. The power needed to cover a set of receiving units is the maximum of the power needed to reach any of them, and grows at least quadratically with the distance to the receiving unit. Consequently, computing a minimum energy routing is NP-hard [2]. Therefore, the energy efficiency of applications depends on efficient routing heuristics. The Minimum Energy Broadcast Problem (MEBP) has attracted intensive research, of which an overview can be found in the survey of Guo and Yang [4]. The most frequently cited algorithm for MEBP is the Broadcast Incremental Power (BIP) algorithm by Wieselthier et al. [8].

Previous works on MEBP heuristics emphasize low time complexity and approximability [2], [6], [7]. In [8], Wieselthier et al. gave an implementation of BIP with $O(|V|^3)$ time complexity when applied to a graph $G = (V, A)$ representing a wireless network. In this work, we present an implementation having $O(|V|^2)$ time complexity by adapting the $O(|A| + |V| \log |V|)$ implementation of Prim’s algorithm using Fibonacci heaps.

Wan et al. [6] together with Klasing et al. [5] showed that, under assumptions commonly made on wireless signal propagation, the approximation ratio of BIP is between $4 + \frac{1}{3}$ and 12.15. Due to a lemma in [6] and Ambühl’s work [1], the currently best upper bound on the approximation ratio of BIP is 6. Retaining the assumptions in [5] and [6], we strengthen the lower bound by giving a sequence of MEBP instances for which the optimal power consumption decreases towards 1, and for which the power consumption of BIP’s solution
increases beyond 4.598. Worst-case instances do not only provide lower bounds on the approximation ratio, but also point out an algorithm’s weakness, and thus suggest directions for future algorithm development.

2 Preliminaries

A problem instance is given by a graph \( G = (V, A) \), a source \( s \in V \), and power requirements (costs) \( c : A \to \mathbb{R} \). The nodes and arcs represent the networking units and potential wireless links, respectively. We define \( c_{vw} = \infty \) for all nodes \( v, w \) where \((v, w) \notin A\).

A solution can be given by an \( s \)-arborescence \( T = (V_T, A_T) \) with node set \( V_T \subseteq V \) and arc set \( A_T \subseteq A \). An \( s \)-arborescence is a directed tree where all arcs are oriented away from \( s \). An \( s \)-arborescence \( T \) induces for every \( v \) a power assignment \( p_v(T) \), which either is 0 or the cost \( c_{vw} \) of the most expensive arc \((v, w) \) leaving \( v \) in \( A_T \). Thereby, the cost of \( T \) is \( p_T = \sum_{v \in V_T} p_v(T) \), and the minimum energy broadcast problem can be formulated as

\[
\text{[MEBP]} \quad \text{Find an } s \text{-arborescence } T \text{ such that } V_T = V \text{ and } p_T \text{ is minimized.}
\]

BIP constructs an \( s \)-arborescence \( T = (V_T, A_T) \) in a way similar to Prim’s construction of a minimum spanning tree. Starting from \( T = (\{s\}, \emptyset) \), BIP evaluates all arcs \((u, v)\) where \( u \in V_T \) and \( v \notin V_T \) by the incremental power \( c_{uv} - p_u(T) \). An arc \((u, v)\) minimizing this difference is selected, and \( v \) and \((u, v)\) are added to \( T \). This is repeated until \( T \) spans \( V \).

3 Improved time complexity of BIP
Our implementation (Tab. 1) follows the implementation of Prim’s algorithm in [3]. It keeps all vertices \( v \in V \setminus V_T \) in a min-priority queue \( Q \) based on a key field containing the minimum incremental cost of adding \( v \) to \( V_T \). The field \( \pi[v] \) contains a node in \( V_T \) to which \( v \) can be linked at cost \( \text{key}[v] \).

In Tab. 1, \( V_T \) and \( A_T \) are represented by \( V \setminus Q \) and \( \{(\pi[v], v) : v \in V_T \setminus \{s\}\} \), respectively. The adjacency list \( \text{Adj}[v] \) contains all nodes \( w \) for which \((v, w) \in A\).

**Theorem 1** The implementation in Tab. 1 has \( O(|V|^2) \) time complexity.

**PROOF.** Follows immediately by observing that the while-loop is iterated \(|V| - 1\) times, that none of the for-loops need more than \(|V| \) iterations each time they are entered, and that each iteration of the for-loops runs in constant time. \( \square \)

For determining a multicast routing that reaches all nodes in a specified destination set, Wieselthier et al. suggested in [7] to apply BIP first, and then omit (“prune”) all arcs not leading to a destination, resulting in the Multicast Incremental Power (MIP) algorithm. Pruning is done by traversing the arborescence upwards from the leaves to the first node that either is the source, or

---

### Table 1. Implementation of BIP

<table>
<thead>
<tr>
<th>BIP ((G = (V, A), s, c))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 for all ( v \in V \setminus {s} )</td>
</tr>
<tr>
<td>2 ( \text{key}[v] \leftarrow c_{sv} )</td>
</tr>
<tr>
<td>3 ( \pi[v] \leftarrow s )</td>
</tr>
<tr>
<td>4 priority queue ( Q \leftarrow V \setminus {s} )</td>
</tr>
<tr>
<td>5 while ( Q \neq \emptyset )</td>
</tr>
<tr>
<td>6 ( v \leftarrow \text{extractMin}(Q) )</td>
</tr>
<tr>
<td>7 for all ( w \in \text{Adj}[v] \cap Q )</td>
</tr>
<tr>
<td>8 if ( c_{vw} &lt; \text{key}[w] )</td>
</tr>
<tr>
<td>9 ( \text{key}[w] \leftarrow c_{vw} )</td>
</tr>
<tr>
<td>10 ( \pi[w] \leftarrow v )</td>
</tr>
<tr>
<td>11 for all ( w \in \text{Adj}[\pi[v]] \cap Q )</td>
</tr>
<tr>
<td>12 if ( (c_{\pi[v]w} - c_{\pi[v]v} &lt; \text{key}[w]) )</td>
</tr>
<tr>
<td>13 ( \text{key}[w] \leftarrow c_{\pi[v]w} - c_{\pi[v]v} )</td>
</tr>
<tr>
<td>14 ( \pi[w] \leftarrow \pi[v] )</td>
</tr>
<tr>
<td>15 return ( T = (V, {(\pi[v], v) : v \in V \setminus {s}}) )</td>
</tr>
</tbody>
</table>
a destination, or a node with more than one child. Traversed arcs and their head nodes are deleted. Next, $p_v(T) \ (\forall v \in V_T)$ can be computed using breadth first search (BFS) in $T$. Since both traversing $T$ and BFS have $O(|V|)$ time complexity, MIP has $O(|V|^2)$ time complexity.

4 A new lower bound on BIP’s approximation ratio

For any $u, v, w \in \mathbb{R}^2$ and $r \in \mathbb{R}$, we denote by

- $uv$ the line segment with end points $u$ and $v$,
- $d_{uv}$ the length of $uv$, i.e. the Euclidean distance between $u$ and $v$,
- $\angle uvw$ the angle between the line segments $uv$ and $vw$ with positive (counter-clockwise) direction from $uv$ to $vw$, that is, the angle for which

$$\cos \angle uvw = \frac{d_{uu}^2 + d_{vw}^2 - d_{uw}^2}{2d_{uu}d_{vw}}, \quad (1)$$

- $C(u, r) = \{ x \in \mathbb{R}^2 : d_{ux}^2 = r^2 \}$ = the circle with radius $r$ centered at $u$.

An important evaluation criterion for algorithms is the performance of the algorithm solution relative to the optimal one. For any instance $I$ of a minimization problem and any algorithm $A$, the performance ratio $\rho_A(I)$ is defined as the cost of the algorithm’s solution divided by the cost of the optimal solution. The supremum $\sup_I \rho_A(I)$ over all possible input instances is called the approximation ratio of $A$, on which a lower bound is given by $\rho_A(I)$ for any instance $I$.

It follows from [1] and [6] that $\sup_I \rho_{BIP}(I) \in [4 + \frac{1}{3}, 6]$, if the supremum is taken over instances where $G$ is complete, $V$ is a finite set of points in $\mathbb{R}^2$, and $c_{uv}$ is proportional to $d_{uv}^2 \ \forall (u, v) \in A$. Since the proportional factor is
irrelevant in our analysis we assume $c_{uv} = d^2_{uv} \forall (u, v) \in A$. For this type of instances, we construct a sequence for which the performance ratio of BIP increases beyond 4.598.

4.1 The best lower bound known from the literature

In [6], Wan et al. gave an instance for which BIP outputs an $s$-arborescence with power consumption arbitrarily close to $4 + \frac{1}{3}$ times the optimal. To the best of our knowledge, this is the best lower bound on the approximation ratio of BIP known to date. In Fig. 1, we depict a slightly modified version of the instance in [6], yielding the same bound. The modification is made in order to prepare for a stronger bound, which through extensions of the instance in Fig. 1 will be derived in subsequent sections.

Our instance contains the nodes $a_0, \ldots, a_4, b_0, \ldots, b_3$, and $z_0, \ldots, z_m$, where $m > 1$ is an integer,

\[ a_j = \left( \cos \left( \frac{(2+j)\pi}{3} \right), \sin \left( \frac{(2+j)\pi}{3} \right) \right) \quad (j = 0, \ldots, 4), \]

\[ b_j = \left( 1 + \frac{2}{\sqrt{3m}} \right) a_j \quad (j = 0, \ldots, 3), \quad z_m = \left( 0, \frac{1}{\sqrt{3}} \right), \quad z_j = \frac{jz_m}{m} \quad (j = 0, 1, \ldots, m). \]

Hence, $z_0 = (0, 0)$, and $z_0, \ldots, z_m$ are uniformly distributed along $z_0 z_m$, and $a_0, \ldots, a_4$ ($b_0, \ldots, b_3$) are positioned on (close to) the unit circle $C(z_0, 1)$. We let the source be $s = z_m$.

The idea in [6] is to make BIP prefer chordal arcs (e.g., $(a_0, a_1)$) to the corresponding radial arcs (e.g., $(z_0, a_1)$), which in [6] is accomplished by perturbing the position of some of the nodes on $C(z_0, 1)$ such that $d_{a_{j-1}, a_j}$ becomes

![Fig. 1. Instance where BIP has performance ratio $4 + \frac{1}{3}$](image)
marginally smaller than \( d_{z_0 a_0} \). We apply the same idea, and the desired effect is obtained by encouraging a marginal power assigned to \( a_0, \ldots, a_3 \) in order to reach \( b_0, \ldots, b_3 \) (not present in the instance of [6]), respectively.

It is readily seen that for the instance in Fig. 1 with sufficiently large \( m \), some optimal arborescence contains arcs \((z_m, z_{m-1}), \ldots, (z_1, z_0), (z_m, a_0), (a_0, b_0), \ldots, (a_3, b_3), (z_0, a_1), \ldots, (z_0, a_4)\), resulting in a power consumption of \( 1 + O(m^{-1}) \). The arborescence \( T' \) produced by BIP contains the arcs \((z_m, z_{m-1}), \ldots, (z_1, z_0), (z_m, a_0), (a_0, b_0), \ldots, (a_3, b_3), (a_0, a_1), \ldots, (a_3, a_4)\), resulting in \( p_{T'} = 4 + \frac{1}{3} + O(m^{-1}) \).

The three terms of \( p_{T'} \) reflect the path \((a_0, \ldots, a_4)\), the arc connecting \( a_0 \) to the source, and arcs of marginal length, respectively. In the following, a better lower bound on the approximation ratio of BIP is derived from similar instances, where:

- the optimal power consumption remains close to 1,
- BIP produces an arborescence containing the path \((a_0, \ldots, a_4)\), and
- BIP connects \( a_0 \) to the source at a cost higher than \( \frac{1}{3} \).

### 4.2 Increasing the cost of connecting \( a_0 \) to the source

Let the positions of nodes \( z_0, a_0, \ldots, a_4 \) be the same as in Sect. 4.1. In order to increase the cost of connecting \( a_0 \), we define a new source \( s_1 \) in a position further away from \( a_0 \). In the new instance \( \mathcal{I}_1 \) shown in Fig. 2, we also:

- Maintain a set \( Z_1 \) of \( m+1 \) nodes uniformly distributed along some curve \( \zeta_1 \) with end points \( s_1 \) and \( z_0 \), thus keeping the optimal power consumption arbitrarily close to 1.
• Keep $a_3$ as a closest neighbor to $a_4$, so that BIP links $a_4$ to $a_3$.

• Keep $a_0$ at least as close to $s_1$ as to any other node in $Z_1$, so that BIP links $a_0$ to $s_1$.

The two last restrictions imply $d_{za_0} \geq d_{s_1a_0}$ and $d_{za_4} \geq 1$ for all $z \in Z_1$, meaning that $\zeta_1$ cannot intersect the interiors of $C(a_0, d_{a_0s_1})$ and $C(a_4, 1)$. Hence we must choose $s_1$ close enough to $a_0$ to make the circles intersect at only one point, which is obtained by letting $d_{a_0s_1} = d_{a_0a_4} - 1 = \sqrt{3} - 1$. Among the two intersection points in $C(a_0, \sqrt{3} - 1) \cap C(z_0, 1)$, we let the source $s_1$ be the point closer to $a_4$. We let $\zeta_1$ be the curve which starts at $s_1$, follows $C(a_0, \sqrt{3} - 1)$ in negative direction to the unique point of intersection of $C(a_0, \sqrt{3} - 1)$ and $C(a_4, 1)$, and from there follows $C(a_4, 1)$ in positive direction to $z_0$. Let $z_m = s_1$, and let the set $Z_1$ consist of $m + 1$ nodes $z_m, \ldots, z_0$ distributed along $\zeta_1$ such that $d_{zm} = \cdots = d_{z_1z_0} = \varepsilon_1$, implying that $\varepsilon_1$ tends to 0 as $m$ grows towards infinity. The node set of instance $\mathcal{I}_1$ is $S_1 = Z_1 \cup \{a_0, \ldots, a_4, b_0, \ldots, b_3, t_1\}$, where $b_j = (1 + 2\varepsilon_1)a_j$, $(j = 0, \ldots, 3)$, and $t_1 = (1 + 2\varepsilon_1)s_1$.

Note that in order to reach $a_0$, the source must at least be assigned power $(\sqrt{3} - 1)^2 = 4 - 2\sqrt{3}$. In the next section, we prove that BIP assigns this power to $s_1$, and that $\mathcal{I}_1$ yields an improved lower bound on the approximation ratio of BIP. However, an even stronger bound is achieved by generalizing the
instance, and we therefore give the proof for a class of instances including $I_1$.

4.3 Instances with performance ratio $> 4.598$

The construction of $I_1$ indicates that $Z_1$ can be extended by adding nodes in the region bounded by $C(z_0, 1)$, $C(a_4, 1)$ and $C(a_0, \sqrt{3} - 1)$. Our idea is to find a sequence of new source locations diverging from $a_0$ and converging to $C(a_4, 1)$, while satisfying the conditions given at the end of Sect. 4.1.

In the following, we construct a sequence $\{I_1, I_2, \ldots\}$ of MEBP-instances having the optimal power consumption converging to 1, and the power consumption of BIP’s solution converging to a number larger than 4.598. Instance $I_i$ is given by a recursive definition of a source $s_i$ and a curve $\zeta_i$ with end points $s_i$ and $z_0$. The basis of this recursion is $s_1$ and $\zeta_1$ introduced in Fig. 2.

For convenient notation, let $s_0 = a_0$. Generalizing the determination of $s_1$, the location of $s_i$ ($i \geq 1$) is the intersection point in $C(s_{i-1}, d_{s_{i-1}a_4} - 1) \cap C(z_0, 1)$ closest to $a_4$ (Fig. 3). The curve $\zeta_i$ follows $C(s_{i-1}, d_{s_{i-1}s_i})$ from $s_i$ until it reaches $\zeta_{i-1}$, after which $\zeta_i$ and $\zeta_{i-1}$ coincide. We let the set $Z_i$ consist of nodes $z_m, \ldots, z_0$, where $z_m = s_i$, distributed along $\zeta_i$ such that $d_{z_m z_{m-1}} = \cdots = d_{z_1 z_0} = \varepsilon_i$, where $m$ is sufficiently large to satisfy $2\varepsilon_i < d_{s_{i-1}s_i}$. To complete the definition of $I_i$, let $b_j = a_j(1 + 2\varepsilon_i)$, ($j = 0, \ldots, 3$), $t_j = s_j(1 + 2\varepsilon_i)$, ($j = 1, \ldots, i$), and let the node set of $I_i$ be $S_i = Z_i \cup \{a_0, \ldots, a_4, b_0, \ldots, b_3, s_1, \ldots, s_i, t_1, \ldots, t_i\}$.

We have $s_0 = (x_0, y_0) = \left(\frac{-1}{2}, \frac{\sqrt{3}}{2}\right)$, and $s_i = (x_i, y_i)$ ($i = 1, 2, \ldots$) is given by:
\[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 = \left(\sqrt{(x_{i-1} - 1)^2 + y_{i-1}^2} - 1\right)^2 \quad i = 1, 2, \ldots (2)\]

\[x_i^2 + y_i^2 = 1 \quad i = 1, 2, \ldots (3)\]

Fig. 3. Instance \(I_3\) with performance ratio \(\rho_{BIP}(I_3) > 4.598\)

Table 2. How BIP processes \(I_i\)

Theorem 2: The performance ratio \(\rho_{BIP}(I_i)\) is \(4 + \sum_{j=1}^{i} c_{s_j, s_{j-1}} + O(m^{-1})\).

Proof.

The optimal total power is \(1 + O(m^{-1})\). We prove that BIP processes \(I_i\) as shown in Tab. 2, yielding total power consumption \(4 + \sum_{j=1}^{i} c_{s_j, s_{j-1}} + O(m^{-1})\).

Since \(2\varepsilon_i < d_{s_i, s_{i-1}}\), Steps 2 and 4 are obvious. Step 5 follows from the fact that adding \(s_{j-1}\) to the tree gives minimum incremental cost, and \(s_j\) is the best choice for linking \(s_{j-1}\).

After the execution of Steps 1-5, assume the for-loop 6-8 has been executed \(j \in \{0, \ldots, 3\}\) times, which means that \(a_1, \ldots, a_{j+1}\) and \(b_0, \ldots, b_j\) have been added. It is then obvious that Step 7 follows since \((a_j, b_j)\) is the only arc adding a new node to the tree with \(O(m^{-2})\) incremental cost, for \(j = 0, \ldots, 3\).

If \(j < 3\), Step 8 follows since the corresponding incremental cost is \(1 - 4\varepsilon_i^2\), whereas all other options of tree augmentation cost at least \(1 - \varepsilon_i^2\). If \(j = 3\), \(a_4\)
is the only node that is not yet reached. In order to show that it is linked to
$a_3$ rather than any of $s_1,s_2,\ldots,s_i$, we prove $c_{s_ja_4} - c_{s_js_j-1} > 1 \forall j = 1,\ldots,i$.

By applying (2)-(3) for $i = 1$ and $i = 2$, we obtain $(x_1,y_1) \approx (0.224,0.975)$,
$(x_2,y_2) \approx (0.455,0.891)$, $c_{s_1a_4} - c_{s_1s_0} \approx 1.02 > 1$ and $c_{s_2a_4} - c_{s_2s_1} \approx 1.03 > 1$.

To prove $c_{s_ia_4} - c_{s_is_i-1} > 1$ for $i > 2$, let $\sigma_i = \angle a_4z_0s_i$ ($i = 0,1,\ldots$), yielding
$\angle a_4s_i-1s_i = \frac{\sigma_i}{2}$ ($i > 0$). As $d_{s_ia_4} > 1$ by construction, we have $\sigma_i > \frac{\pi}{2}$. Thus
$\cos \angle a_4s_i-1s_i < \frac{\sqrt{3}}{2}$. By (1),
$$
\cos \angle a_4s_{i-1}s_i = \frac{c_{s_{i-1}a_4} + c_{s_{i-1}s_i} - c_{s_is_i}}{2d_{s_{i-1}a_4}d_{s_{i-1}s_i}}.
$$
This yields
$$
c_{s_ia_4} - c_{s_is_i-1} = c_{s_{i-1}a_4} - 2 \cos \angle a_4s_{i-1}s_i \cdot d_{s_{i-1}a_4}d_{s_{i-1}s_i} >
$$
$$
> c_{s_{i-1}a_4} - \sqrt{3} \cdot d_{s_{i-1}a_4}d_{s_{i-1}s_i} =
$$
$$
= c_{s_{i-1}a_4} - \sqrt{3} \cdot d_{s_{i-1}a_4} \left(d_{s_{i-1}a_4} - 1\right) = \left(1 - \sqrt{3}\right)c_{s_{i-1}a_4} + \sqrt{3}d_{s_{i-1}a_4}.
$$

The polynomial $(1 - \sqrt{3})x^2 + \sqrt{3}x - 1$ attains positive values between its
zeroes, which are at $x = 1$ and $x \approx 1.366$. By construction, we have $1 < c_{s_ia_4} \leq c_{s_2a_4} \approx 1.09$. Hence $c_{s_ia_4} - c_{s_is_i-1} > 1$ ($i = 3,4,\ldots$), and Step 8 follows.

The proof is completed by observing that the output of Table 2 has a total
power of $4 + \sum_{j=1}^{\infty} c_{s_j,s_j-1} + O(m^{-1})$. □

**Corollary 3** The approximation ratio of BIP is at least
$$
4 + \sum_{j=1}^{\infty} c_{s_j,s_j-1} > 4 + \sum_{j=1}^{3} c_{s_j,s_j-1} > 4.598.
$$

**PROOF.** Solving (2)-(3) numerically for $i = 1,\ldots,3$ gives the result. □

A simple analysis shows that to a precision of three decimals, 4.598 is the
best achievable bound by the sequence of instances. It is seen from Fig. 3 that
\[ \sum_{j=i+1}^{\infty} c_{s_j s_{j-1}} < c_{s_i s_{\infty}}, \text{ where } s_{\infty} = \lim_{i \to \infty} s_i = \left( \frac{1}{2}, \frac{\sqrt{3}}{2} \right). \] Solving (2)-(3) also for \( i = 4 \) yields \( 4 + \sum_{j=1}^{4} c_{s_j s_{j-1}} = 4.5983 \pm 0.5 \cdot 10^{-5} \) and \( c_{s_4 s_{\infty}} < 0.5 \cdot 10^{-5}. \)

4.4 Extension to sweep

Wieselthier et al. [7] introduced the local search method sweep for MEBP. Given a spanning arborescence \( T \), sweep searches for an arc \((v,w) \in A_T\) and a node \( u \in V \setminus \{v,w\} \) such that \( p_v(T) = c_{vw}, \ p_u(T) \geq c_{uw}, \) and \( u \) is not a descendant of \( w \) in \( T \). Such a combination implies that \((v,w)\) can be replaced by \((u,w)\) in \( A_T \), thus reducing the power at \( v \) (unless \( c_{vw} = c_{wv'} \) for some other child \( w' \) of \( v \)).

If \( T \) is the arborescence produced by applying BIP to \( I_i \), it is easily checked that for all feasible sweep moves, we have \((v,w) \in \{(z_m, z_{m-1}), \ldots, (z_1, z_0)\}\). Note that \((v,w) \in \{(s_i, s_{i-1}), \ldots, (s_2, s_1)\}\) with \( u = a_0 \) is infeasible since \( a_0 \) descends from \( s_1, \ldots, s_{i-1} \). Since \( p_{z_j}(T) = O(m^{-2}) \) \((j = 1, \ldots, m - 1)\), the power reduction obtainable by sweep is only \( O(m^{-1}) \). Thus Corollary 3 also applies when BIP and sweep are run sequentially.

5 Conclusions

We have proposed an implementation of BIP/MIP with \( O(|V|^2) \) time complexity, and demonstrated that the approximation ratio of BIP is larger than 4.598. The latter holds also if the local improvement method sweep is applied to the solution produced by BIP.
References


