Power Efficient Routing Trees for Ad Hoc Wireless Networks Using Directional Antenna [⋆]

Fei Dai, Qing Dai, and Jie Wu

Department of Computer Science and Engineering
Florida Atlantic University, Boca Raton, Florida 33431

Abstract

In ad hoc wireless networks, nodes are typically powered by batteries. Therefore saving energy has become a very important objective, and different algorithms have been proposed to achieve power efficiency during the routing process. Directional antenna has been used to further decrease transmission energy as well as to reduce interference. In this paper, we discuss five algorithms for routing tree construction that take advantage of directional antenna, i.e, Reverse-Cone-Pairwise (RCP), Simple-Linear (SL), Linear-Insertion (LI), Linear-Insertion-Pairwise (LIP), and a traditional approximation algorithm for the travelling salesman problem (TSP). Their performances are compared through a simulation study.

Key words: ad hoc wireless network, directional antenna, energy-efficient routing, transmission power, travelling salesman problem (TSP)
PACS:

1 Introduction: background and related work

Energy efficiency is an important consideration for ad hoc wireless networks, where nodes are typically powered by batteries. It is directly correlated with network longevity and connectivity, therefore affecting network throughput. Among all different components of power consumption, transmission cost appears to dominate, compared to receiving cost and computation cost. It has

[⋆] The work was supported in part by a grant from Motorola Inc. and NSF grants CCR 0329741, ANI 10073736, and EIA 0139806.
Email addresses: `fdai@fau.edu` (Fei Dai), `qdai_jfgao@hotmail.com` (Qing Dai), `jie@cse.fau.edu` (Jie Wu).

```
    Algorithm I: Reverse-Cone-Pairwise (RCP)
1   $unreachedNodes \leftarrow$ {all nodes except $sourceNode$}
2   $reachedNodes \leftarrow \{sourceNode\}$
3   $edgesInGraph \leftarrow \emptyset$
4   $totalCost \leftarrow 0$
5   while $unreachedNodes \neq \emptyset$
6       for $i \leftarrow 0$ to $size[reachedNodes] - 1$
7           for $j \leftarrow 0$ to $size[unreachedNodes] - 1$
8               do find $minNode$ $j$ in $unreachedNodes$ with
                        minimum $incrementalCost(i, j)$
9     $reachedNodes \leftarrow reachedNodes \cup minNode$
10    $unreachedNodes \leftarrow unreachedNodes - minNode$
11    $edgesInGraph \leftarrow edgesInGraph \cup \{edge(i, j)\}$
12    $totalCost \leftarrow totalCost + incrementalCost(i, j)$
```

Fig. 1. Algorithm I: Reverse-Cone-Pairwise (RCP).

been shown that the power threshold p for a source node to reach its destination node is positively correlated to the distance between them, and can usually be expressed as $p = r^{\alpha} + c$ for some constants α and c [1], where r is the distance between the two nodes, and α is between 2 and 4. In previous studies, different metrics have been used. Some measure the overall energy consumption of the network, while some others try to extend lifespan of individual nodes. The Broadcast Incremental Power algorithm (BIP) is a centralized algorithm attempting to minimize the overall energy in route determination [2]. It is similar to Prim's Minimum Spanning Tree (MST) algorithm [3], in that at any time all reached nodes form a single-rooted tree. Each step adds the node with the minimum incremental cost, calculated as the additional energy for it to be reached by any node within the tree, either by increasing power of a transmitting node, or by making a non-transmitting node transmit at a specific power level. It has already been proved that BIP has a constant approximation ratio of between 6 and 12, compared to the optimal solution [4]. On the other hand, some power-aware routing approaches select routes that avoid nodes with low remaining power, which can be either absolute or relative power level. Different metrics may well lead to different algorithms, but they can also be combined to achieve a balance, thus optimizing overall energy consumption without depleting crucial nodes [5,6].

Recently, the use of directional antenna was proposed in order to reduce interference and to further increase power efficiency, because ideally, power consumed in the case of directional antenna is only

$$r^{\alpha}\,\frac{\theta}{2\,\pi}\,,$$

where θ is the beam angle. In one of the session-based studies, two algorithms were proposed: RB-BIP (Reduce Beam BIP) and D-BIP (Directional BIP).

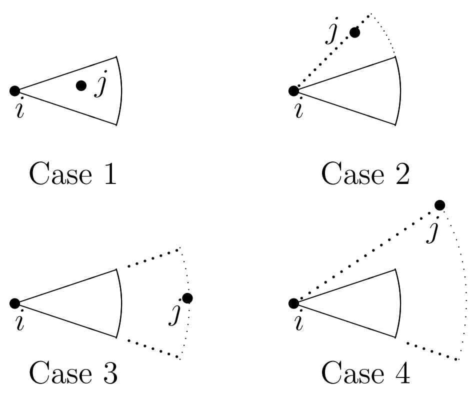


Fig. 2. Incremental transmission cost.

The former simply adds a beam-reducing step to the original BIP algorithm, so that each node can now transmit at its smallest possible angle. The latter incorporates the use of directional antenna at each step of the tree construction, i.e, a node in the tree could also increase its current transmission beam angle or shift the existing beam to reach a new destination node, in addition to increasing its transmission power, whichever gives the lowest incremental cost [5].

In our study, we try to find different centralized routing algorithms for broadcasting with the use of directional antenna. Four new algorithms are proposed to create a routing tree in ad hoc networks. In the Reverse-Cone-Pairwise (RCP) algorithm, the transmission beam of each sender is adjustable to achieve a good balance between the number of covered receivers and the transmission power. This algorithm is a refinement to the original D-BIP scheme. The difference is that RCP employs a new transmission beam adjustment scheme, which incurs less incremental cost when expanding the beam to cover an additional receiver. The remaining three algorithms, Simple-Linear (SI), Linear-Insertion (LI), and Linear-Insertion-Pairwise (LIP), form a linear chain to connect all nodes in the network, where each sender uses the minimal beam angle to reach its next hop in the chain. These algorithms have lower computation cost than RCP, and is appropriate for ad hoc networks using very narrow transmission beams. We also investigate a traditional travelling salesman (TSP) algorithm, which can also form a chain for broadcasting. All those schemes are evaluated via a simulation study, and their energy efficiency is compared with that of D-BIP.

The remainder of this paper is organized as follows: five algorithms are introduced in Section 2, including four new algorithms and one existing algorithm for comparison. We then simulate their performance with randomly generated networks in Section 3, and discuss their applications and potentials in Section 4.

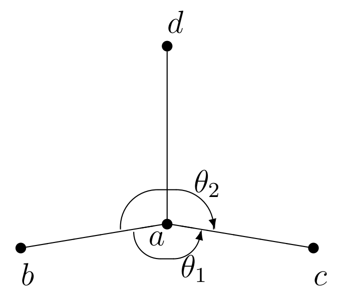


Fig. 3. Illustration of Reverse-Cone method.

2 Algorithms

Our objective is to find routing algorithms that are power efficient using directional antenna. In this study, we assume one beam for each node, and $\alpha = 2$ and $c = 0$ to calculate the transmission energy. Here we propose four algorithms, assuming global knowledge of node locations, adjustable transmission range and adjustable beam angle. In all three algorithms, we placed a constraint that each sender forms only one transmission beam to reach its receivers. Simultaneously forming several beams is possible, but requires extra hardware support. Multiple beams can be emulated via consequent transmissions in several directions. However, switching transmission directions causes energy and penalty. Therefore, it is better to keep the beam direction as long as possible, instead of changing direction frequently [7]. In addition, allowing multiple beams renders the original problem a trivial one, which is equivalent to constructing a MST when the minimum beam angle is very small.

The first algorithm is a refinement of D-BIP [5], where the node with the minimum incremental cost is added at each step, while taking the adjustable antenna beam width into consideration (see Figure 1). The actual beam angle that is used has to exceed a minimum beam angle *minAngle*. All nodes that have already been reached can act as transmitting node, and all the non-tree nodes are potential destination nodes. Each possible transmitting-destination node pair is examined to determine the minimum power and beam angle needed to add a new destination node. When a transmitting node is adding a new destination node, if the new node does not fall into its current transmission beam, it can either shift or expand its previous beam to cover the new node. The incremental power is determined by subtracting the energy of the previous transmission beam from the new power, calculated as

$$incrementalCost(i, j) = (p' \times \theta' - p \times \theta)/(2 \times \pi),$$

where p' and θ' represent the new transmission power and beam angle, respectively, while p and θ are the previous power and beam angle. This process

```
**MinBeam** ($srcNode$, $destNode$)
calculate $destAngle$
insert $destAngle$ into $angleList$
$maxCone \leftarrow 0$
$minBeam \leftarrow (beamStart \leftarrow 0, beamEnd \leftarrow 0)$
**for** $i \leftarrow 0$ to $size[angleList]$
  $coneAngle \leftarrow angleList[i+1] - angleList[i]$
  **if** $coneAngle > maxCone$
    $maxCone \leftarrow coneAngle$
    $minBeam \leftarrow (angleList[i+1], angleList[i])$
**return** $minBeam$
```

Fig. 4. Reverse-Cone method to find the minimum beam.

is demonstrated in Figure 2, where i and j are the source and destination nodes, respectively, and the cone represents the current transmission beam. In case 1, the destination node d could be enclosed within the current beam, including through a beam shift. It will be covered without any incremental cost. Otherwise, either the transmission beam has to be expanded (case 2) or transmission power needs to be increased (case 3), or both (case 4) for d to be reached. It could therefore be very costly.

During this process, it is desirable to use the minimum transmission beam angle whenever possible. A simple heuristic method to calculate the new beam is to keep the start and end point of the previous beam, and to expand either end that leads to a smaller increase of the overall beam span. We will see that this heuristic does not always provide the optimal beam angle. An example is illustrated in Figure 3. At first, the source node a is transmitting to two downstream neighbors b and c with a beam angle θ_1. When a new destination node d joins, and when θ_1 is close to π, neither expanding beam to b-c-d nor to c-b-d provides the minimum beam. On the other hand, b-d-c is the optimal solution in this case with a beam angle θ_2. The Reverse-Cone method (Figure 4) is developed to calculate the minimum beam span. Instead of merely keeping the start and end point of the previous beam, each node keeps angle positions, calculated as the radius of a destination node from itself, of all its destination nodes in a sorted list *angleList*. The 2π circle around a transmitting node is then divided into several cones, each defined by the two adjacent neighbors in the *angleList*. Whenever adding a new destination node, the transmitting node will first insert the angle location of the new node into its *angleList*. The *angleList* is traversed to find the largest cone. The minimum new beam to cover all the nodes in *angleList* would be the reverse of this largest cone.

For the previous example in Figure 3, node a first adds nodes b and c in its sorted *angleList*. Of the two cones, $c - b$ counterclockwise is larger than $b - c$, therefore the reverse of cone $c - b$, i.e, cone $b - c$, or θ_1, is selected as the minimum beam. When node a adds a new destination d, d is then inserted into a's angle list, which now becomes $d - b - c$. After traversal of the new

```
Algorithm II: Simple Linear (SL)
1   $unreachedNodes \leftarrow \{$all nodes except $sourceNode\}$
2   $reachedNodes \leftarrow \{sourceNode\}$
3   $listHead \leftarrow sourceNode$
4   $totalCost \leftarrow 0$
5   while $unreachedNodes \neq \emptyset$
6     for $j \leftarrow 1$ to $size[unreachedNodes] - 1$
7       do find $minNode$ with minimum cost $minCost$
8     $reachedNodes \leftarrow reachedNodes \cup minNode$
9     $listHead \leftarrow minNode$
10    $unreachedNodes \leftarrow unreachedNodes - minNode$
11    $totalCost \leftarrow totalCost + minCost$
```

Fig. 5. Algorithm II: Simple Linear (SL).

angleList, $b - c$ is found to be the largest cone, therefore the reverse of it, $c - d - b$, or θ_2 will be used as the new transmission beam.

The RCP algorithm generally provides good performance, except in the situations when the previous transmission beam was small, adding a new node at a distant angle position can be very expensive. Three other heuristic algorithms are therefore developed: Simple-Linear (SL), Linear-Insertion (LI), and Linear-Insertion-Pairwise (LIP). In all of them, each transmitting node always uses the minimum beam angle *minAngle* to reach exactly one downstream node, and there is only one downstream destination node for each transmitting node. As a result, a linear chain will be formed step by step, starting from the source node. Initially, the *reachedNodes* set includes *sourceNode* only, and the *unreachedNodes* set includes all other nodes. In SL (see Figure 5), initially the source node is the *listHead*, then at each step, a *minNode* is determined as an *unreached* node closest to the current *listHead*, and is added to the *reachedNodes* set to become the new *listHead*. This *listHead* is the only possible transmitting node to reach the next new node, until the *unreachedNodes* set is empty.

In LI (Figure 6), an additional backtrack step is included when each node is added to the *reachedNodes* set. During the backtrack, after the *minNode* is determined, it is first tested for possible insertions into each position between any two adjacent nodes within the existing linear chain. The *insertionCost*, the incremental cost for inserting the new node, is calculated to check whether any insertion causes a saving of energy, compared to directly adding the *minNode* as the new *listHead*. If so, an insertion will take place where the *insertionCost* is the minimum. In this case, the previous *listHead* remains unchanged. If not, *minNode* would be attached as the new *listHead*. In the example in Figure 7, s represents the *sourceNode*, d_1 through d_6 are the destination nodes to be reached. The Linear-Insertion algorithm would add d_6 between s and d_1 in the backtrack process, which leads to a lower overall

```
**Algorithm III: Linear-Insertion (LI)**
$unreachedNodes \leftarrow$ {all nodes except $sourceNode$}
$reachedNodes \leftarrow \{sourceNode\}$
$listHead \leftarrow sourceNode$
$totalCost \leftarrow 0$
while $unreachedNodes \neq \emptyset$
  for $j \leftarrow 0$ to $size[unreachedNodes] - 1$
    do find $minNode$ with minimun cost $minCost$
  for $k \leftarrow 0$ to $size[reachedNodes] - 2$
    do insert $minNode$ after $reachedNode[k]$
       find minimum $InsertCost$
  if $InsertCost < minCost$
    do $minCost \leftarrow minInsertCost$
       insert $minNode$
  else
    do $listHead \leftarrow minNode$
  $reachedNodes \leftarrow reachedNodes \cup minNode$
  $unreachedNodes \leftarrow unreachedNodes - minNode$
  $totalCost \leftarrow totalCost + minCost$
```

Fig. 6. Algorithm III: Linear Insertion (LI).

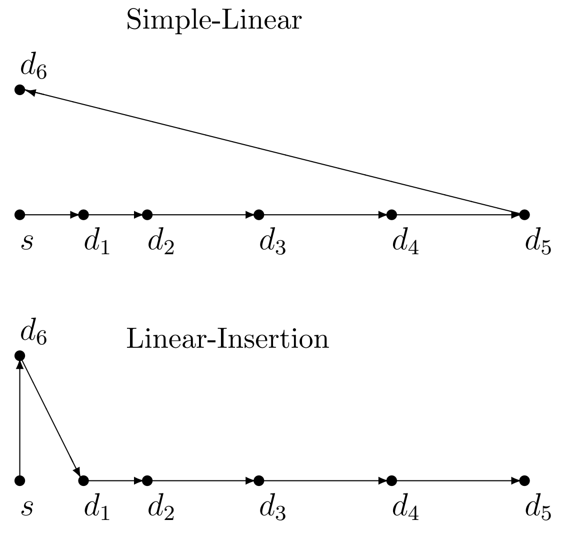


Fig. 7. Illustration of Linear-Insertion.

energy cost. While LI seems to be a better solution than SL, unfortunately, it does not always outperform SL due to its heuristic nature (examples not shown).

LIP (Figure 8) takes one further step beyond LI. In LIP, the incremental cost for inserting every node in the *unreachedNode* set to the existing chain is computed. The *minNode* with the minimum incremental cost, instead of the

```
    Algorithm IV: Linear-Insertion-Pairwise (LIP)
1   $unreachedNodes \leftarrow \{$all nodes except $sourceNode\}$
2   $reachedNodes \leftarrow \{sourceNode\}$
3   $totalCost \leftarrow 0$
4   while $unreachedNodes \neq \emptyset$
5     for $j \leftarrow 0$ to $size[unreachedNodes] - 1$
6       for $k \leftarrow 0$ to $size[reachedNodes] - 1$
7         do insert node $i$ after $reachedNode[k]$
8            find $minNode$ in $unreachedNodes$ with the
9                 minimum insertion cost $minCost$
10    insert $minNode$
11    $reachedNodes \leftarrow reachedNodes \cup minNode$
12    $unreachedNodes \leftarrow unreachedNodes - minNode$
13    $totalCost \leftarrow totalCost + minCost$
```

Fig. 8. Algorithm IV: Linear Insertion Pairwise (LIP).

minimum distance to the *listHead*, is removed from the *unreachedNode* set and inserted into the *reachedNode* chain. To simplify the description, we view attaching a node j to the *listHead* and make j the new *listHead* as a special case of insertion, and use $Cost(listHead, j)$ as the incremental cost in this case. In each single step, LIP can find a *minNode* with lower *minCost* than LI. Again, it is a heuristic approach, and does not always outperform LI.

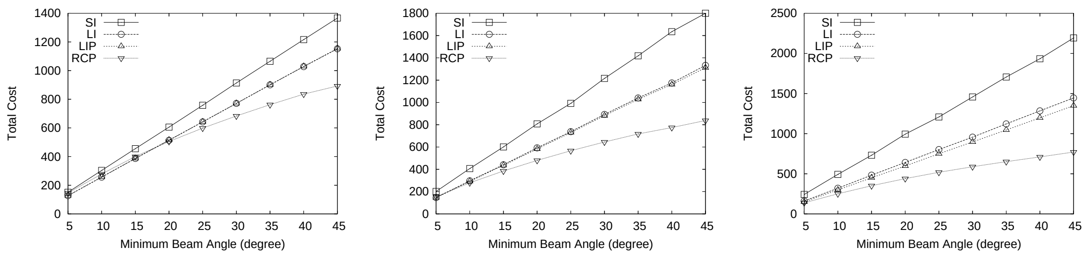
The problem of forming a minimum path to visit all nodes has been known as the travelling salesman problem (TSP), which is NP-complete. TSP has an approximation solution in geographic graphs [8], where each node is placed in a 2-D area, and the cost of an edge (i, j) is proportional to the distance of the two end nodes i and j. The TSP approximation algorithm (Figure 9) consists of two phases. The first phase (lines 1–11) constructs a MST using Prim's algorithm [3]. The second phase (lines 13–21) constructs a chain by taking a pre-order walk of the MST. This algorithm has an approximation ratio of 2. That is, the total cost of the path is at most twice that in an optimal solution. Although this approximation ratio does not apply in non-geographic graphs, we can still use it as a benchmark in performance comparisons.

The complexity of the five algorithms are as follows: SI and LI have the similar computation complexity of $\Theta(n^2)$. LIP and TSP have a higher complexity of $\Theta(n^3)$. RCP has the highest complexity of $\Theta(n^4)$, with the Reverse-Cone method. To evaluate the relative performance of the three algorithms, simulations are performed on random networks. The result is shown in Section 3.

```
**Algorithm V: Travelling-Salesman (TSP)**
$unreachedNodes \leftarrow$ {all nodes except $sourceNode$}
$reachedNodes \leftarrow \{sourceNode\}$
$children[i] \leftarrow \emptyset$ for all nodes $i$
while $unreachedNodes \neq \emptyset$
  for $i \leftarrow 0$ to $size[reachedNodes] - 1$
    for $j \leftarrow 0$ to $size[unreachedNodes] - 1$
      do find $minNode$ in $unreachedNodes$ with the
              minimum $Cost(i, j)$
  $reachedNodes \leftarrow reachedNodes \cup minNode$
  $unreachedNodes \leftarrow unreachedNodes - minNode$
  $children[i] \leftarrow children[i] \cup \{minNode\}$

$listHead \leftarrow \emptyset$
$stack \leftarrow \{sourceNode\}$
$totalCost \leftarrow 0$
while $stack \neq \emptyset$
  do $j \leftarrow pop(statck)$
     $push(stack, children[j])$
     $listHead \leftarrow j$
     if $listHead \neq \emptyset$
        do $totalCost \leftarrow totalCost + Cost(listHead, j)$
```

Fig. 9. Algorithm IV: Travelling Salesman Approximation



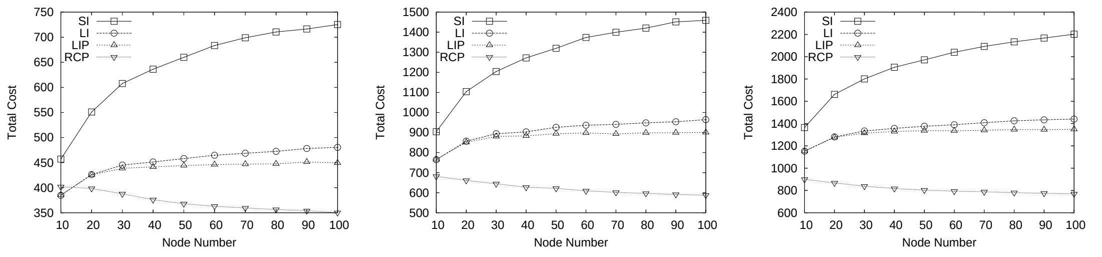
(a) 10 nodes (b) 30 nodes (c) 100 nodes

Fig. 10. Simulation results with the minimum beam angle varying from $\pi/36$ (5°) to $\pi/4$ (45°).

3 Simulation Results

We generate random network instances, and the four algorithms are used to obtain routing graphs, respectively. Their transmission costs are calculated and compared with each other.

Our results show that in most occasions, RCP produces routing solutions with good energy-efficiency. Between SL, LI, and LIP, LI has a better performance



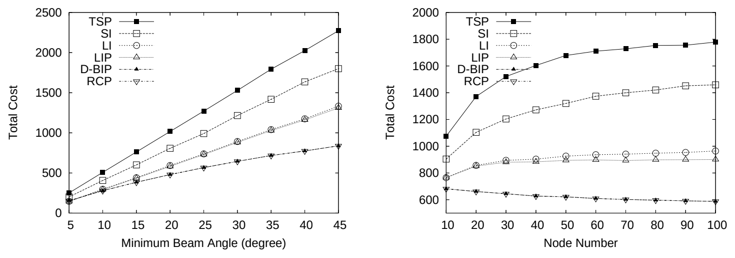
(a) Minimal beam angle $\pi/12$ (15°) (b) Minimal beam angle $\pi/6$ (30°) (c) Minimal beam angle $\pi/4$ (45°)

Fig. 11. Simulation results with number of nodes from 10 to 100.

than SL, and LIP is slightly better than LI. In simulation with different network density, we observed that when node distribution is sparse, LI and LIP outperform RCP when the minimal beam angle is relatively small. In the example shown in Figure 10, we set the number of nodes from 10 to 100 in the same size of area. As the minimal beam angle increases gradually, we can see that energy costs of SL, LI, and LIP routing increase with the minimal angle linearly. This is because in those algorithms, the route construction process is not affected by the minimal angle, and the same routing graph will always be produced just as omnidirectional antenna is provided. The final cost is simply calculated by multiplying the original overall cost by a factor of $minAngle/2\pi$. The result of RCP is close to SI when the minimal angle is small, which makes sense since in this case RCP would most likely generate the same routing graph as SL. When the minimal angle increases to a certain point, RCP will outperform SL, and eventually outperform LI and LIP, when shifting or expanding current beam shows advantage. The difference between LI and LIP is trivial in sparse networks with 10 or 30 nodes. This difference becomes obvious in dense networks, where LIP can find a better $minNode$ in each step among a large pool of candidates.

Simulations are also performed to measure the overall cost over the number of nodes. In a fixed area, the costs of SL, LI and LIP increment along with the increase of nodes, but the cost of RCP decreases, except when the node number and minimal beam angle are very small (Figure 11). It seems that in most cases, RCP has the best scalability among the three.

To compare the performance of our three algorithms, especially RCP, with that of D-BIP, a simplified version of D-BIP is implemented. As before, we consider only the transmission cost, which is a function of the distance. As expected, RCP shows better energy efficiency than D-BIP (Figure 12). However, the difference among RCP and D-BIP is very small. The same figure also shows the performance of the traditional TSP algorithm. It is obvious that all algorithms proposed in this paper outperform TSP.



(a) 30 nodes

(b) Minimal beam angle $\pi/6$ (30°)

Fig. 12. Simulation results of proposed schemes compared with TSP and D-BIP.

Nevertheless, all four algorithms described above are greedy algorithms and heuristic in nature, with none being optimal. In fact, we can find special network instances between SI, LI, LIP or RCP, where one outperforms the other two(examples not shown). From the simulation result, it seems that when nodes are relatively sparse and the minimal beam angle is small, LI is the best choice. Not only does it give the lowest routing cost, but its complexity is lower than RCP also. The routing cost of LIP is close to that of LI, but LIP has a higher computation complexity than LI. When the minimal angle increases to a certain extent or when the network is dense, expanding current beam to include additional destination nodes would be more cost efficient, and at this point, RCP provides the best performance.

4 Conclusion

The problem of broadcasting a message to the entire ad hoc network with minimum energy is NP-complete. Using directional antennas can further reduce the energy consumption, but also makes the problem more complicated. In this paper, we have proposed four heuristic algorithms that construct a broadcast tree based on global information. Those algorithms have different performance in terms of reducing the broadcast cost and incur different computation cost. A simulation study has been conducted to provide guidelines for tradeoffs under different network situations. These algorithms are also compared with two existing algorithms in terms of their energy efficiency. Currently, we are further analyzing their performance statistically. In the future, it will be interesting to know the approximation ratio of the those algorithms.

References

[1] V. Rodoplu, T. Meng, Minimum energy mobile wireless networks, IEEE Journal on Selected Areas in Communications 17 (8).

[2] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, On the construction of energy-efficient broadcast and multicast trees in wireless networks, in: Proceedings of IEEE INFOCOM, 2000, pp. 585–594.

[3] R. C. Prim, Shortest connection networks and some generalizations, Bell System Technical Journal 36 (1957) 1389–1401.

[4] P. J. Wan, G. Calinescu, S. Y. Li, O. Frieder, Minimum-energy broadcasting in static ad hoc wireless networks, Wireless Networks 8 (2002) 607–617.

[5] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, Energy-limited wireless networking with directional antennas: the case of session-based multicasting, in: Proceedings of IEEE INFOCOM, 2002, pp. 190 –199.

[6] J. Wu, F. Dai, M. Gao, I. Stojmenovic, On calculating power-aware connected dominating set for efficient routing in ad hoc wireless networks, IEEE/KICS Journal of Communications and Networks 4 (2002) 59–70.

[7] A. Spyropoulos, C. S. Raghavendra, Energy efficient communications in ad hoc networks using directional antennas, in: Proceedings of IEEE INFOCOM, 2002.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 2nd Edition, The MIT Press, 2001.