

Power Assignment For Symmetric Communication In Wireless Sensor Networks *

Joongseok Park Sartaj Sahni
Computer & Information Science & Engineering
University of Florida
{jpark, sahani}@cise.ufl.edu

February 12, 2007

Abstract

We show that two incremental power heuristics for power assignment in a wireless sensor network have approximation ratio 2. Enhancements to these heuristics are proposed. It is shown that these enhancements do not reduce the approximation ratio of the considered incremental power heuristics. However, experiments conducted by us indicate that the proposed enhancements, reduce the power cost of the assignment on average. Further, the two-edge switch enhancement yields a power-cost reduction (relative to using minimum cost spanning trees) that is, on average, twice as much as obtainable from any of the heuristics proposed earlier.

Keywords: Power assignment, symmetric connectivity, wireless sensor networks, approximation algorithm.

1 Introduction

We consider the problem of assigning power levels to the sensors in a wireless sensor network so that the total power assigned is minimum and the communication graph of the sensor network has a path composed solely of bidirectional edges¹. between every pair of sensors. We assume that the sensors are battery operated and are deployed in an environment in which it is impractical (or at least undesirable) to recharge/replace the battery of a sensor (e.g., in a battle field or at the

*This research was supported, in part, by the National Science Foundation under grant ITR-0326155

¹When a directed graph contains the directed edges (u, v) and (v, u) , we say that (u, v) is bidirectional. A bidirectional edge may be modeled as an undirected edge. Hence, we use the terms bidirectional and undirected interchangeably.

bottom of an ocean). Hence, energy conservation is paramount. Although sensors use energy to sense as well as to communicate among themselves, our focus here is on the conservation of energy spent in communication activities. Depending on the application, inter sensor communication may be modeled as a unicast, multicast or broadcast.

We assume the omnidirectional antenna model in which the n -sensor wireless network is represented as a complete weighted directed graph G that has n vertices/nodes (we use the terms sensor, vertex and node interchangeably). Each node of G represents a sensor of the wireless network. The weight $w(i, j)$ of the directed edge (i, j) is the minimum power level at which sensor i must transmit if its message is to be received by sensor j . Using an omnidirectional antenna, sensor i can transmit to sensors j_1, j_2, \dots, j_k , by setting its power level to

$$\max\{w(i, j_q) | 1 \leq q \leq k\}$$

In the most common model used for power attenuation in wireless broadcast, signal power attenuates at the rate a/r^d , where a is a media dependent constant, r is the distance from the signal source, and d is another constant between 2 and 4 [22]. So, for this model, $w(i, j) = w(j, i) = c * r(i, j)^d$, where $r(i, j)$ is the Euclidean distance between nodes i and j and c is a constant. In practice, however, this nice relationship between $w(i, j)$ and $r(i, j)$ may not apply. This may, for example, be due to obstructions between the nodes that may cause the attenuation to be larger than predicted. Also, the transmission properties of the media may be asymmetric resulting in $w(i, j) \neq w(j, i)$. When $w(i, j) \neq w(j, i)$ for some (i, j) , we say that the *power requirements are asymmetric*. When $w(i, j) = w(j, i)$ for all i and j , the power requirements are symmetric.

In an effort to conserve energy, each sensor u adjusts its power level to a value $p(u)$ such that the *power cost* $p(G) = \sum_u p(u)$ is minimum and the subgraph H of G formed by (directed) edges (u, v) such that $w(u, v) \leq p(u)$ is strongly connected. The resultant subgraph is called a *minimum power strongly connected topology*. Note that with the set power levels, sensor u can do a single-hop transmission to every sensor v for which (u, v) is an edge of H . Since H is strongly connected, every sensor can communicate with every other sensor using a multi-hop transmission. Figure 1(a) shows a network with 5 sensors A through E. Each circle represents the transmission range of the sensor at its center given this sensor's current power level. The example 5-sensor network may be represented by a 5-vertex complete digraph with $w(i, j)$ being the minimum $p(i)$

needed to transmit from i to j . Figure 1(b) shows the subgraph H of this complete digraph with the property that for every edge (u, v) in H , $p(u) \geq w(u, v)$. Notice that while sensor E can do a single-hop transmission to sensor D , sensor D cannot do a similar transmission to E . This is because $p(E) \geq w(E, D) = w(D, E) > p(D)$. Although H is strongly connected, it is asymmetric ((E, D) is an edge of H but (D, E) is not). This asymmetry in the connectivity of H poses a problem in applications where the single-hop transmission protocol requires the receiver to send an acknowledgment to its one-hop neighbor who sent the message. So, while D can receive from E , in Figure 1(b), D cannot send a one-hop acknowledgment of this receipt back to E . To overcome this problem, we require that H contain a spanning connected subgraph comprised solely of bidirectional edges. By using only the edges in this spanning connected subgraph, all single-hop transmissions can be acknowledged by the receiver. Figure 1(c) shows a connected spanning subgraph of Figure 1(b) that is comprised solely of bidirectional edges. The bidirectional edges are shown as undirected edges in this figure.

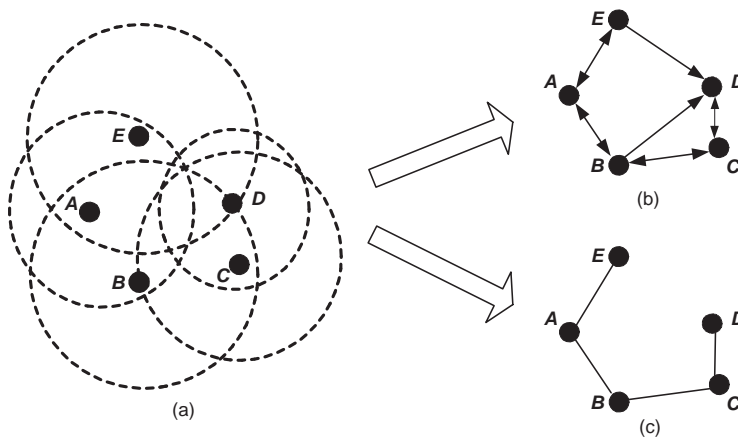


Figure 1: The connectivity model (each node has the different transmission power level): (a) Sensors and their transmission ranges (b) An asymmetric communication graph and (c) A symmetric subgraph of (b)

A formal statement of the problem we study in this paper is given below.

Minimum Power Symmetric Connectivity Problem (*MPSC*)

Input: A complete weighted undirected graph G .

Output: A power assignment $p()$ to the vertices of G so that the power cost $p(G) = \sum_u p(u)$ is minimum and the subgraph of G comprised of bidirectional (undirected) edges (u, v) such that $p(u) \geq w(u, v)$ and $p(v) \geq w(u, v)$ (equivalently, $\min\{p(u), p(v)\} \geq w(u, v)$) is a connected spanning subgraph of G .

In Section 2, we review prior research related to the MPSC problem. In Section 3, we prove that the incremental power heuristics of [17, 4] have an approximation ratio of 2 matching the approximation ratio of the MST (minimum spanning tree) heuristic of [3, 19, 4, 10]. Enhancements of these heuristics as well as to the MST heuristic are presented in Section 4. In Section 5, we present experimental results that show that our refinements result in improved performance.

2 Related Work

The minimum energy broadcast tree problem (MEBT) is related to MPSC. In MEBT, we are to assign power levels to the nodes of G so that $\sum_u p(u)$ is minimum and the graph comprised of directed edges (u, v) such that $p(u) \geq w(u, v)$ contains a directed tree, which is rooted at a specified source node s and which includes all vertices of G . MEBT is NP-hard, because it is a generalization of the connected dominating set problem, which is known to be NP-hard [8]. In fact, MEBT cannot be approximated in polynomial time within a factor $(1 - \epsilon)\Delta$, where ϵ is any small positive constant and Δ is the maximal degree of a vertex, unless $NP \subseteq DTIME[n^{O(\log \log n)}]$ [9]. Clementi et al. [6] show that MEBT is NP-hard when edge weights equal the Euclidean distance between the nodes (sensors). Wieselthier et al. [15, 16] consider the case when $w(i, j) = c * r^d$. They develop a fast MEBT algorithm for the case of networks that have 3 nodes (a source broadcasting to 2 other nodes). For larger networks, they propose a recursive algorithm that “makes more than 51,000 calls” when there are 10 destination nodes. Since the proposed recursive algorithm is impractical for large networks, several heuristics are proposed. Four of the proposed heuristics are greedy heuristics that scale well as the network size increases. Wan et al. [14] provide a theoretical analysis of the performance of three of the greedy heuristics proposed in [15, 16]. They show that two of the proposed heuristics (link-based minimum spanning tree and broadcast incremental power) have constant approximation ratios and a third (shortest path spanning tree) has an approximation ratio that is at least $n/2$ (these approximation ratios are for the case

$$w(i, j) = c * r^d.$$

Let MPAC be the MPSC problem with the relaxation that the subgraph of G defined by the assigned power levels contain a strongly connected spanning subgraph. So it is possible that (u, v) is in the subgraph ($p(u) \geq w(u, v)$) and (v, u) is not in the subgraph ($p(v) < w(v, u) = w(u, v)$). Chen and Huang [3] have shown that MPAC is NP-hard for general graphs. The proof of [3] applies also to MPSC. An approximation algorithm that is based on minimum cost spanning trees and that has approximation ratio 2 also is developed in [3]. This approximation algorithm appears to have been rediscovered many times by several researchers [19, 4, 10]. Kirousis et al. [10] show that MPAC is NP-hard when the edge weights are equal to distances in 3-dimensional space (E^3) and Clementi et al. [7] prove this for distances in 2-dimensional space (E^2).

Calinescu et al. [19] show that for a given G , the total power in the optimal solution for MPAC can be half that for the optimal solution for MPSC when the distances are in E^2 . They develop also two approximation algorithms for MPSC. One is a fully polynomial time approximation scheme whose approximation ratio is $1 + \ln 2 + \epsilon$ and the other is a greedy algorithm whose approximation ratio is $15/8$. Althaus et al. [17] establish lower approximation ratios of $5/3 + \epsilon$ and $11/6$, respectively, for the two approximation algorithms of [19]. An integer linear programming formulation for MPSC is developed in [17]. Using this formulation, MPSC instances with up to 40 sensors can be solved in about 1 hour of time on a 600MHz PC. [17] asserts that the $5/3 + \epsilon$ algorithm isn't practical and that the $11/6$ algorithm, which is known as the greedy fork contraction algorithm, doesn't perform as well on real data as an extended minimum spanning tree algorithm EFS (edge and fork switching) that begins with a minimum cost spanning tree and iteratively reduces the power cost of this spanning tree. In each iteration, EFS finds the maximum reduction in power possible by switching a tree edge and a non-tree edge as well as by switching a fork (two non-tree edges that share a vertex) and two tree edges. The switch must leave behind a spanning tree. If the maximum reduction is positive, the switch is made and we proceed to the next iteration. Otherwise, EFS terminates. [17] also shows that approximation of MPSC with asymmetric power requirements, is NP-hard for any approximation ratio $(1 - \epsilon) \ln n$, where n is the number of nodes and $\epsilon > 0$ is any constant. Cheng et al. [4] show that MPSC is NP-hard with edge weights equal to distances in E^2 . They propose an incremental power greedy heuristic

for MPSC and demonstrate experimentally that this greedy heuristic performs better than the MST-based heuristic whose approximation ratio is 2.

3 Analysis of Incremental Power Heuristics

Two incremental power heuristics have been proposed to construct minimal energy/power broadcast trees [17, 4, 5, 15, 16]. Both heuristics are suitable for the MEBT and MPSC problems. We describe these two heuristics in the context of the MPSC problem. Conceptually, the two heuristics are identical to the well known Kruskal and Prim algorithms to construct minimum cost spanning trees [12]. In both heuristics we construct a spanning tree of the given graph G by starting with $TE = \emptyset$ and adding edges to TE , one at a time, until TE becomes a spanning tree of G . At any time in the tree construction, the current power level for each vertex u is $\max\{w(u, v) | (u, v) \in TE\}$. The next edge for inclusion into TE is the one that increases the sum of power levels the least and that creates no cycle in TE . When the spanning tree construction is complete, the $p(u)$ s are such that the constructed spanning tree is a subgraph of the communication graph of G defined by the power levels $p(*)$. Hence, these power levels give us a solution to MPSC. Let IPK be the heuristic that results when edge selection is done using Kruskal's strategy and IPP be the one that results when Prim's strategy is used.

Figures 2 and 3 specify the IPK and IPP heuristics. Let $p(TE)$ be the sum of the $p(u)$ s. For edges eligible for selection in each iteration of the **for** loop, $c(u, v)$ gives the incremental cost of adding (u, v) to TE . All references to cost in these heuristics is to $c(*, *)$. It is easy to see that following each iteration of the **for** loop, $p(u) = \max\{w(u, v) | (u, v) \in TE\}$. The resemblance to Kruskal's and Prim's minimum cost spanning tree algorithms is evident.

Figure 4 shows an example for *IPP*. Figure 4(a) shows the sensor network with edge weights equal to the transmission power needed. The number outside a vertex is its current $p()$ value. The weight for each missing edge is ∞ . Figure 4(b) shows the initial edge costs (twice the edge weights) and the initial TV (shaded vertex). This is the configuration at the start of the first iteration of the **for** loop. Each G_i , $i < 8$ gives the edge costs, $p()$, TV (shaded vertices) and TE (solid edges) at the start of the i th iteration of the **for** loop. G_8 gives the final values.

Theorem 1 *IPK and IPP have approximation ratio 2 for MPSC.*

Heuristic $IPK(G)$ {

Step 1: [Initialize]

$TE = \emptyset;$
 $p(u) = 0; 1 \leq u \leq n;$
 $c(u, v) = 2 * w(u, v);$ for all (u, v) in $G;$

Step 2: [Construct spanning tree]

for $i = 1$ **to** $n - 1$ **do** {
 Let $(u, v) \notin TE$ be a least cost edge such that $TE \cup \{(u, v)\}$ is acyclic.
 $TE = TE \cup \{(u, v)\};$
 $p(v) = \max\{p(v), w(u, v)\};$
 $p(u) = \max\{p(u), w(u, v)\};$
 $c(x, y) = \max\{0, w(x, y) - p(x)\} + \max\{0, w(x, y) - p(y)\}$ for $x \in \{u, v\};$
}

}

Figure 2: IPK heuristic for MPSC

Heuristic $IPP(G)$ {

Step 1: [Initialize]

$TV = \{1\};$
 $TE = \emptyset;$
 $p(u) = 0; 1 \leq u \leq n;$
 $c(u, v) = 2 * w(u, v);$ for all (u, v) in $G;$

Step 2: [Construct spanning tree]

for $i = 1$ **to** $n - 1$ **do** {
 Let (u, v) be a least-cost edge such that $u \in TV$ and $v \notin TV.$
 $TV = TV \cup \{v\};$
 $TE = TE \cup \{(u, v)\};$
 $p(v) = w(u, v);$
 $p(u) = \max\{p(u), w(u, v)\};$
 $c(x, y) = w(x, y) + \max\{0, w(x, y) - p(x)\}$ for all $x \in \{u, v\}$ and $y \notin TV;$
}

}

Figure 3: IPP heuristic for MPSC

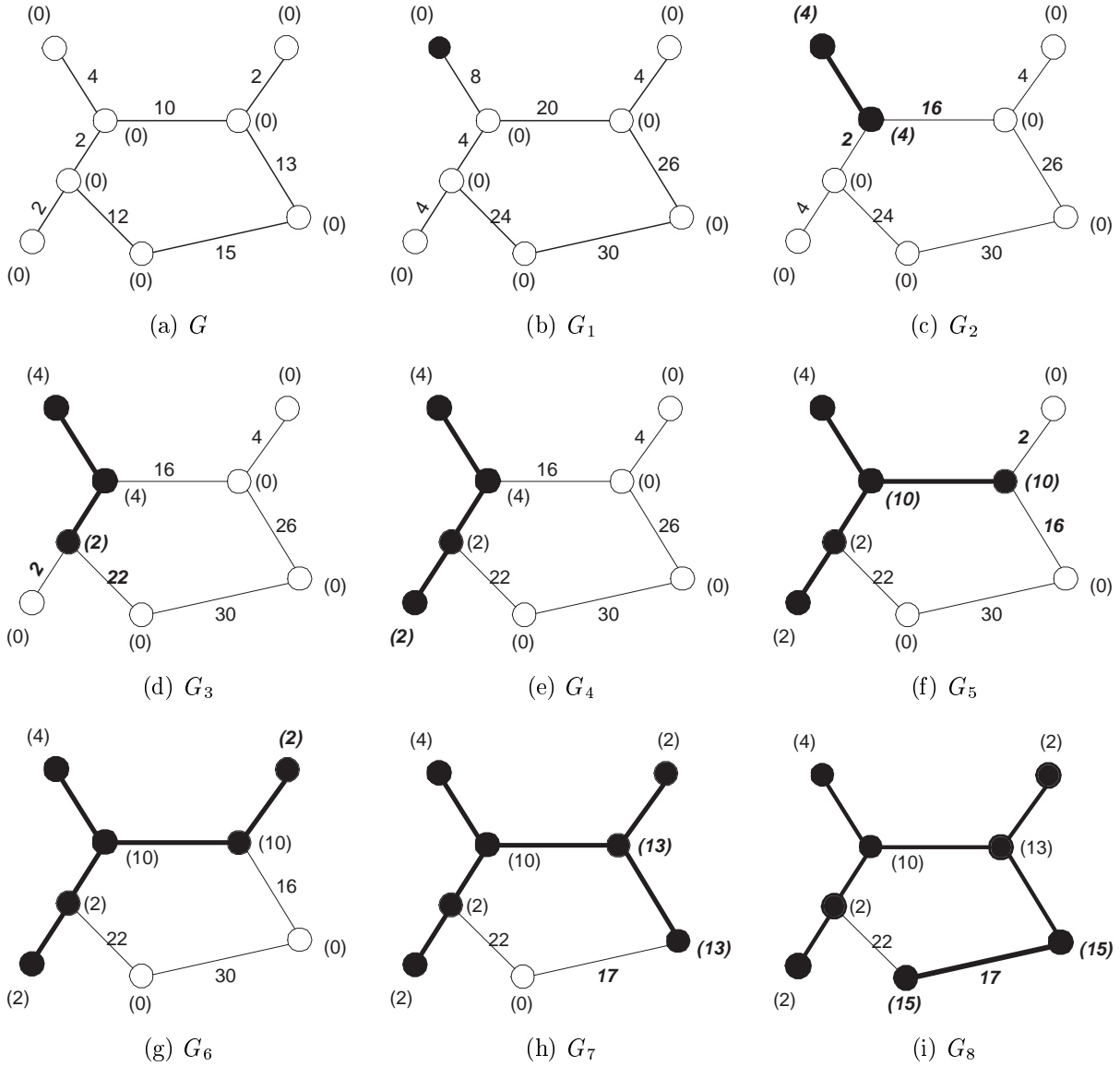


Figure 4: Example for *IPP*

Proof Consider either IPK or IPP (the proof is identical for both). Let G_i be an undirected complete n -vertex graph in which the edge costs are the $c(*, *)$ values at the start of the i th iteration of the **for** loop. Let G_n be this graph at the end of the last iteration of the **for** loop and let TE_i be TE at the start of the i th iteration of the **for** loop. Note that G_1 has $c(u, v) = 2 * w(u, v)$, where $w(u, v)$ is the power need to transmit from u to v (equivalently, from v to u) and $TE_1 = \emptyset$. Let $c(G_i)$ be the cost of the minimum cost spanning tree for G_i subject to the constraint that the spanning tree contain all edges in TE_i . In computing $c(G_i)$, the cost of an edge $(u, v) \in TE_i$ is

its cost at the time it was selected for inclusion in TE and the cost of an edge $(u, v) \notin TE_i$ is its current $c(u, v)$ value. Note that $c(G_n)$ is the sum of $p(*)$ values when the heuristic terminates. This is so as, TE_n is a spanning tree and the sum of edge costs, where the cost of an edge is its $c(*, *)$ value at the time it was added to TE is the sum of the incremental power costs, which is the sum of the final $p(*)$ values.

From the correctness proof of Kruskal's and Prim's algorithms (see [12], for example), it follows that the edge selected for inclusion in TE in iteration i of the **for** loop is in the (constrained) minimum spanning tree of G_i . This together with the observation that the cost of edges eligible for selection in remaining iterations is no more in G_{i+1} than in G_i implies that $c(G_{i+1}) \leq c(G_i)$. Hence, $c(G_n) \leq c(G_1)$.

Note that $c(G_1) = 2w(MST)$, where MST is the minimum cost spanning tree for G_1 and $w(MST)$ is the sum of weights ($w(*, *)$, not $c(*, *)$ values) of the edges in MST . In [3, 19, 10], it is shown that $w(MST) < p(OPT)$, where OPT is an optimal power assignment and $p(OPT)$ is the sum of the power levels in OPT . So, we have

$$c(G_n) \leq c(G_1) = 2w(MST) < 2p(OPT)$$

Since $c(G_n)$ is the sum of the power levels when the heuristic terminates, the heuristic has approximation ratio 2. ■

Theorem 2 *The bound of Theorem 1 is tight.*

Proof Figure 5(a) shows a sensor network. The edges show the permissible one-hop transmissions together with the required transmission power. Note that the weight assignments do not correspond to ideal E^2 distances. As noted in Section 1, the ideal situation almost never arises in practice because of media heterogeneity. Using the symmetric communication graph (which is a spanning tree of Figure 5(a)) of Figure 5(b) results in an optimal power assignment. In this power assignment, the power level of each node on the top row is set to ϵ and that for each node on the bottom row is set to 1. If the total number of nodes is $2n$ (n in each row of nodes), the cost of the optimal power assignment is $n(1 + \epsilon)$. Using a suitable tie breaker, both IPK and IPP construct the spanning tree shown in Figure 5(c). This spanning tree corresponds to a power assignment

of ϵ to each node on the lower row; the assignment for the top row (left to right) is $1, 1.5 - \epsilon/2, 1.75 - 3\epsilon/4, 1.875 - 7\epsilon/8, \dots$. The cost of this power assignment is

$$\begin{aligned} & 1 + (n-1)(2-\epsilon) - (1-\epsilon) \sum_{i=1}^{n-1} 1/2^i - (1-\epsilon)/2^{n-1} \\ &= 2n - 1 + \epsilon - (1-\epsilon) \left[1/2^{n-1} + \sum_{i=1}^{n-1} 1/2^i \right] \end{aligned}$$

As n gets large, the ratio of the cost of the IPK and IPP power assignment to the optimal one approaches $2/(1+\epsilon)$, which approaches 2 as ϵ becomes small. \blacksquare

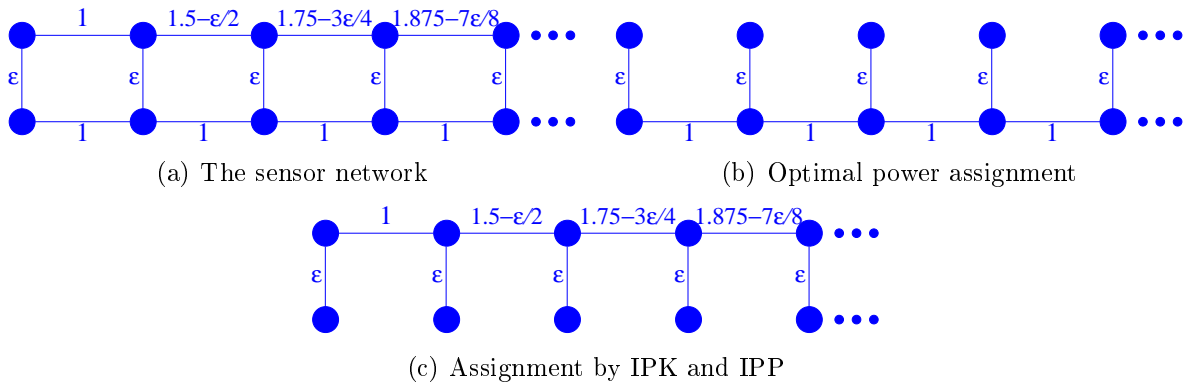


Figure 5: Example for Theorem 2

Theorem 3 Let $p(IPK)$ and $p(IPP)$, respectively, be the sum of the power assignments obtained by IPK and IPP. Let $p(MST)$ be this quantity for the minimum cost spanning tree. $0.5 \leq p(X)/p(MST) \leq 2$ for $X \in \{IPK, IPP\}$. Also, these bounds are tight.

Proof [3, 4, 10] show that $p(MST) \leq 2p(OPT)$, where OPT is the optimal power assignment. From this and Theorem 1, it follows that

$$p(OPT) \leq p(MST) \leq 2p(OPT)$$

$$p(OPT) \leq p(X) \leq 2p(OPT)$$

So, $0.5 \leq p(X)/p(MST) \leq 2$.

To see that these bounds are tight, consider the sensor network of Figure 5(a). Using a suitable tie breaker, the minimum cost spanning tree produced by Kruskal's and Prim's algorithm is that

shown in Figure 6(a). Its power cost is $n(1 + \epsilon) + 1 - \epsilon$, where n is the number of vertices in the top row. The spanning tree generated by X is shown in Figure 5(c). Its power cost approaches $2(n + 1) - 2$ as $n \rightarrow \infty$. So, as n gets large, $p(X)/p(MST)$ approaches 2.

Now consider the network of Figure 6(b). Figure 6(c) shows a possible minimum cost spanning tree obtained by Kruskal’s and Prim’s algorithm. This results in assigning a power level of 2 to all vertices except 2. So, $p(MST)$ is $4n - 4 + 2\epsilon$. Heuristic X , with a suitable tie breaker, produces the spanning tree shown in Figure 6(d). This results in a power assignment of ϵ to each top-row vertex and 2 to each bottom-row vertex. So, $p(X) = n(2 + \epsilon)$. By choosing n and ϵ appropriately, $p(X)/p(MST)$ can be made as close to 0.5 as desired. ■

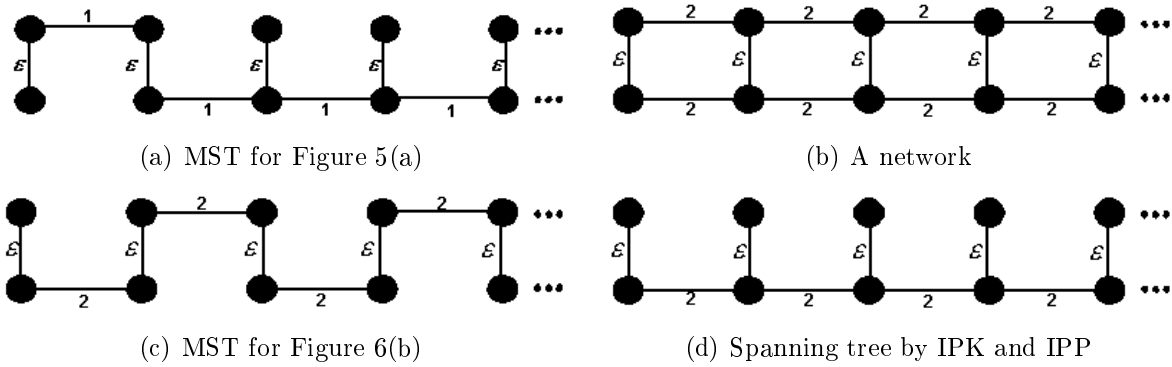


Figure 6: Example for Theorem 3

4 Enhancements

In this section, we describe three strategies to reduce the cost $p(T)$ of the spanning tree T . Although these strategies do not reduce the approximation ratio of the MST , IPK and IPP heuristics, they result in spanning trees that have lower $p(T)$ value on average.

4.1 Sweep Method

Performing a sweep to improve the quality of a spanning tree has been done in the past in the context of the MEBT problem [15, 16, 11]. We propose an adaptation of this method to the MPSC problem. The sweep method of [15, 16, 11] isn’t directly applicable to MPSC because in MEBT we are dealing with asymmetric communication while in MPSC, we have symmetric

- Step 1:** Let v be the nearest not yet tried non-child descendent of u that can be reached using u 's current power level $p(u)$ (i.e., $p(u) \geq w(u, v)$).
- Step 2:** If there is no such v , the visit terminates.
- Step 3:** Move v together with its subtrees so that v is a child of u . Let T' be the new tree.
- Step 4:** Set $p(v) = \max\{w(v, x) | x \text{ is a parent or child of } v\}$.
- Step 5:** Let y be the former parent of v .
Set $p(y) = \max\{w(y, x) | x \text{ is a parent or child of } y\}$.
- Step 6:** If $p(T') < p(T)$ then set $T = T'$.
- Step 7:** Go to Step 1.

Figure 7: Visiting a node u during a sweep

communication. In a sweep, we start by selecting a vertex v of the spanning tree T as the tree root. Following this selection, child-parent relationships are uniquely determined. We perform a level-order traversal [12] of T beginning at the root v . When a node u is visited during this traversal, we perform the steps shown in Figure 7.

As an example, consider the tree T of Figure 8(a). Let u be the root, A , of this tree. Suppose that $p(A) \geq w(A, D)$. So, D is selected as vertex v in Step 1. To get T' , we move D together with its subtrees so that D becomes a child of A . This gives us T' as shown in Figure 8(b). When the power levels are adjusted as in Steps 4 and 5, symmetric communication using T' is possible. If $p(T') < p(T)$, we continue with T' as the working tree. Otherwise, we continue with T . Regardless, during the current visit, D cannot again be selected as the v vertex in Step 1. For an n -vertex spanning tree, the complexity of the visit operation is $O(n)$. In a sweep, each vertex is visited once. So, the complexity of a sweep is $O(n^2)$.

4.2 Single Edge Switch (ES1a and ES1b)

Figures 9 and 10 give the edge switching heuristics ES1a and ES1b. In heuristic ES1a, the edge f is an edge on the unique cycle in $T \cup (u, v)$. In particular, f may be (u, v) . To determine the power of $T - \{f\}$ we need to first adjust the power level of the nodes u, v and the end points of f so that these nodes have the minimum power needed to do a single-hop transmission to their

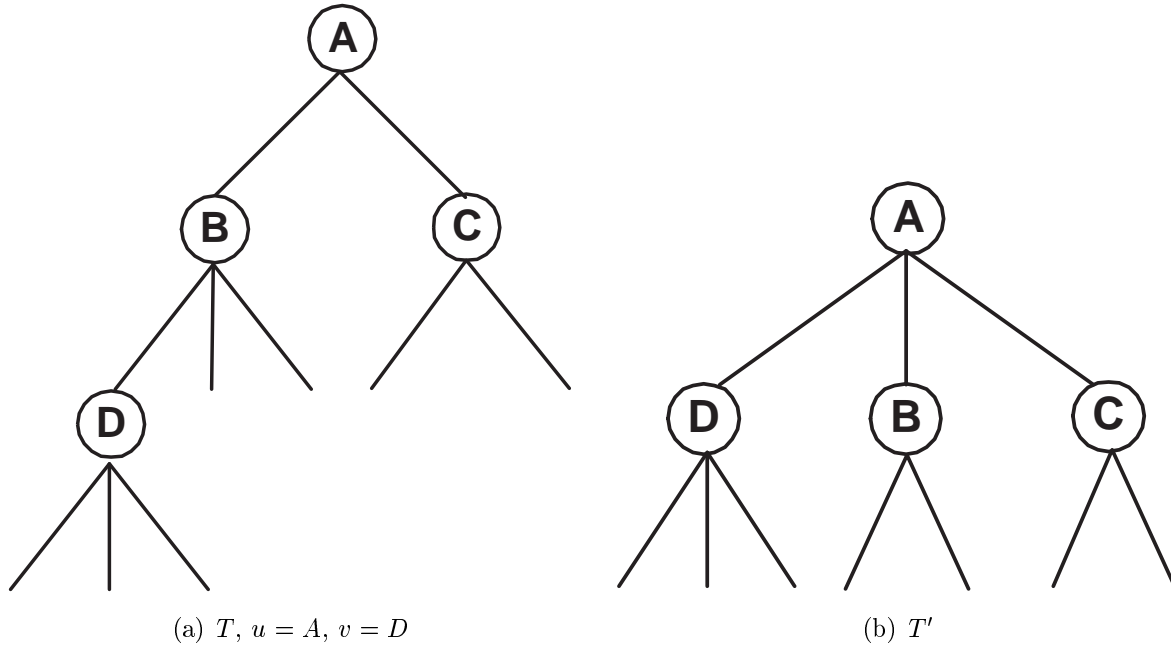


Figure 8: The sweep method

tree neighbors. Note that since (u, v) is a candidate for f , the minimum power of $T - \{f\}$ is no more than that of the tree at the start of the **for** loop iteration. In ES1b, f is an edge that joins the two connected components of $T - \{(u, v)\}$. In particular, f could be (u, v) . Once again to compute the power of $T \cup \{f\}$, the power level of the nodes u, v and the end points of f need to be adjusted as in the case of ES1a.

```

F = edges not in T;
for each edge  $(u, v) \in F$  do
   $T = T \cup \{(u, v)\}$ ;
  Let  $f$  be such that  $T - \{f\}$  is a spanning tree
  with minimum total power;
   $T = T - \{f\}$ ;

```

Figure 9: Heuristic ES1a

Figure 11 shows an example for one iteration of the **for** loop of ES1a. Figure 11(a) shows the tree T at the start of the iteration. The numbers outside a node (e.g., (58) outside $v1$) give the power level of the node. The power cost of this tree is 336. Numbers outside edges are edge weights. $(u, v) = (v4, v7)$ is added to T . This results in the cycle $v4, v7, v1, v6, v4$. The choices for

$F =$ edges in T ;
for each edge $(u, v) \in F$ **do**
 $T = T - \{(u, v)\}$;
 Let f be such that $T \cup \{f\}$ is a spanning tree
 with minimum total power;
 $T = T \cup \{f\}$;
end do

Figure 10: Heuristic ES1b

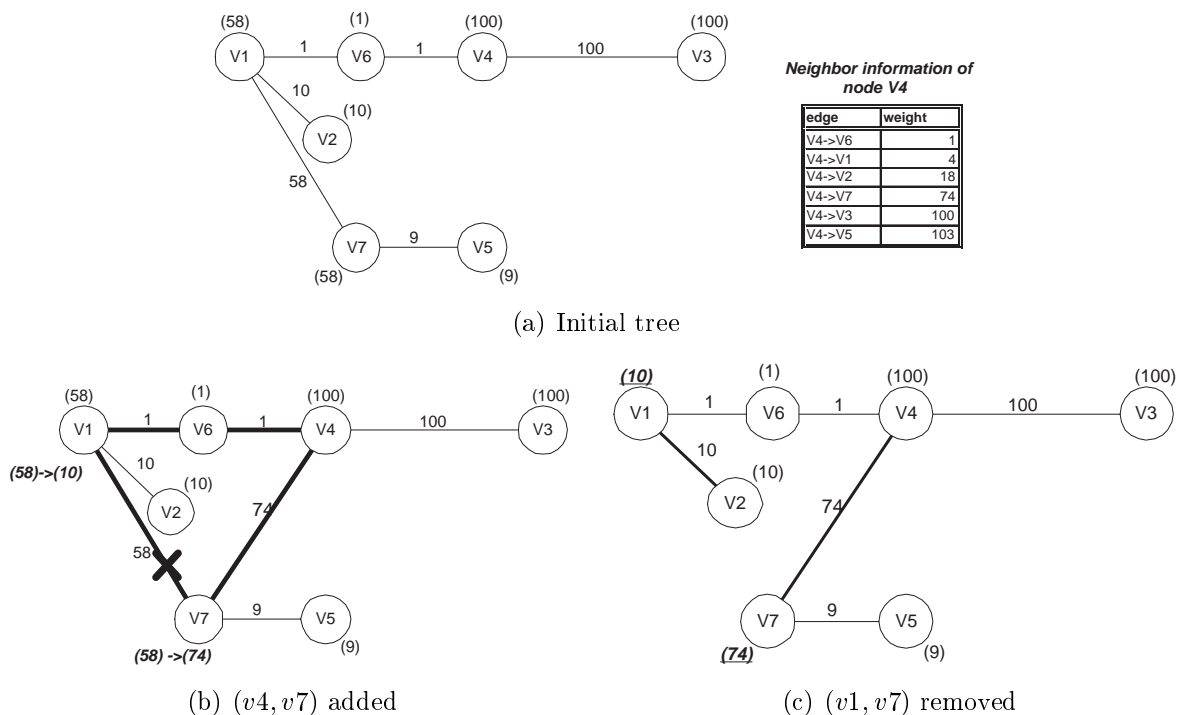


Figure 11: One iteration of the **for** loop of ES1a

f are $(v4, v7)$, $(v7, v1)$, $(v1, v6)$ and $(v6, v4)$. $f = (v1, v7)$ results in $p(v1)$ reducing from 58 to 10 and $p(v7)$ increasing from 58 to 74. The remaining power levels are unchanged. The net reduction in tree power is 32. No other choice of f gives a larger reduction. So, we choose $f = (v1, v7)$ and the new T is as shown in Figure 11(c). This completes one iteration of the **for** loop. In this iteration, $p(T)$ was reduced from 336 to 304.

Figure 12 shows a single iteration of the **for** loop of ES1b. $(u, v) = (v1, v7) \in T$ for this iteration. The edge f that reconnects $T - \{(u, v)\}$ and results in minimum total power is $(v4, v7)$. The result of switching edges $(v1, v7)$ and $(v4, v7)$ is a reduction of 32 in tree power.

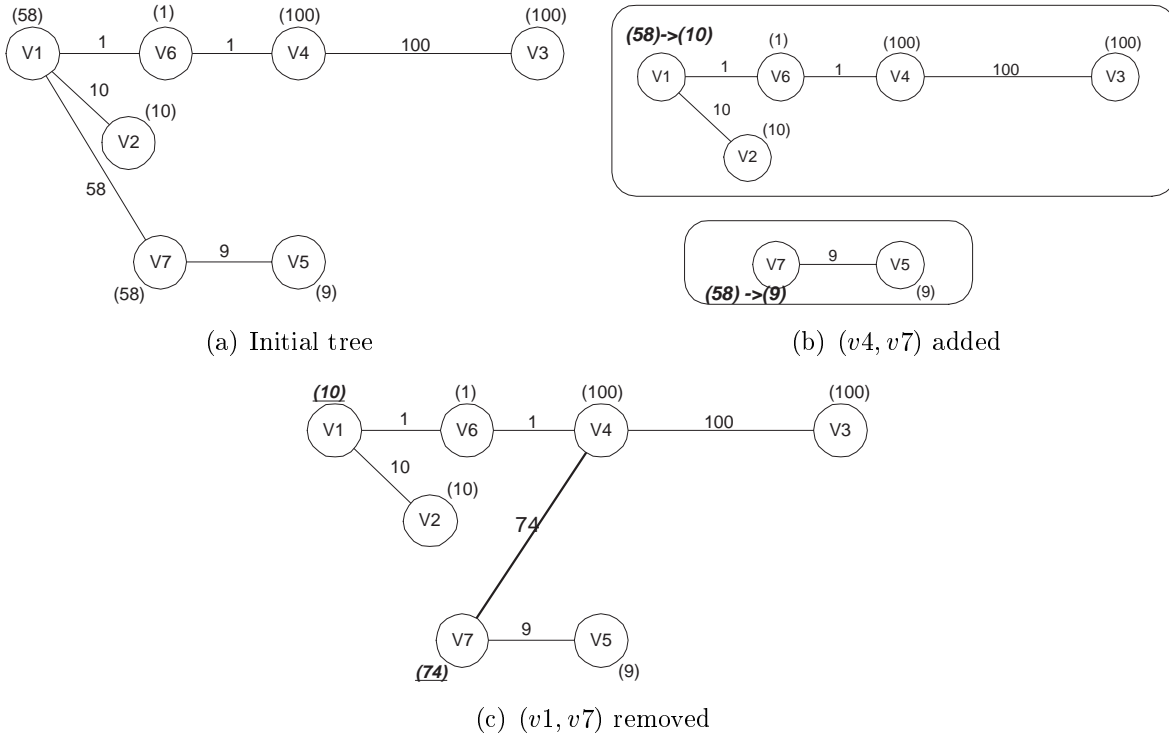


Figure 12: One iteration of the **for** loop of ES1b

ES1b, which is the edge switching heuristic of [20], is simpler than the ES heuristic of [17]. Let (e_1, e_2) be a pair of edges such that e_1 is in the tree and e_2 is not. This edge pair defines a *legal swap* iff deleting e_1 from the tree and adding e_2 results in a spanning tree. In each iteration of ES, an edge pair that defines a legal swap that results in a min power cost spanning tree is determined. If the edge swap reduces power cost, the swap is made and we proceed to the next iteration. Otherwise, the algorithm terminates.

Let e be the total number of edges. The complexity of both ES1a and ES1b is $O(n^2e)$.

4.3 Double Edge Switch (ES2)

Double edge switching is like ES. However, in each iteration we find 4 edges (e_1, e_2, e_3, e_4) such that the first 2 are in the tree and the last 2 are not. The 4 edges define a legal swap iff replacing the first two by the last two results in a spanning tree. As before, a legal swap that results in the minimum power cost spanning tree is determined. The swap is done iff it results in a power reduction. The process continues until no further reduction in power cost is obtainable by a legal

swap. We note that ES2 generalizes the two-edge switching used in the EFS heuristic of [17]. In EFS, the edges e_3 and e_4 are required to form a fork (i.e., the two edges must share a vertex). The complexity of ES2 is $O(n^3e^2)$.

We conclude this section by proving our earlier claim that none of the proposed enhancement methods reduces the approximation ratio of either *IPK* or *IPP*.

Theorem 4 *The approximation ratios of IPK and IPP remain 2 when these heuristics are enhanced by any or all of the methods sweep, single edge switch and double edge switch.*

Proof Follows from Theorem 2 and the observation that none of the stated methods is able to reduce power cost when started with the *IPK* and *IPP* spanning tree of Figure 5(c). ■

5 Experimental Results

The *MST*, *IPK* and *IPP* heuristics together with their enhancements were programmed in C++ and their relative performance compared using random graphs. *MST* was used as the benchmark heuristic for comparison purposes. Our random graphs were generated in a manner similar to that used in [17] to generate test graphs. We experimented with networks that have $n \in \{10, 50, 100\}$ nodes. For each n , we generated 50 random graphs. Each random graph was generated by randomly assigning the n nodes to points in a 10000×10000 grid. We computed $w(i, j) = r(i, j)^d$, where $r(i, j)$ is the Euclidean distance between nodes i and j , and d was set to 2.

Figure 13(a) gives the average percent reduction in power cost realized using either *IPK* or *IPP* over the power cost resulting from *MST*. This average reduction in power cost is between 2% and 3%. Also, *IPP* results in better power assignments than obtained by *IPK*.

Figure 13(b) gives the average percent reduction in power cost realized by the sweep enhancement of *MST* (*MST_SW*), *IPK* (*IPK_SW*) and *IPP* (*IPP_SW*) over the power cost resulting from *MST* and Figures 13(c)–(e), respectively, do this for the ES1a, ES1b and ES2 enhancements of *MST*, *IPK* and *IPP*. Figures 15 and 16 give this average power-cost reduction data together with the minimum and maximum power-cost reductions and the standard deviations for the cases $n = 50$ and 100, respectively. Figure 14 shows the average power cost of the generated power

assignments. This figure has separate bar charts for each of *MST*, *IPK* and *IPP* together with their enhancements.

As can be seen from the provided data, the sweep enhancements of *MST*, *IPK* and *IPP*, on average, result in a reduction in power cost between 3.9% and 5% relative to the power cost from *MST*. The ES1a and ES1b enhancements are about equally effective and both are superior to the sweep enhancement. These enhancements improve upon the *MST* power cost by between 5.2% and 5.7%. However, the greatest power-cost reductions relative to *MST* come when ES2 is employed. Now, the average reduction ranges from 12.5% to 14.2%. The maximum reduction observed on our test data was 24.5%. Note that since the approximation ratio for *MST* is 2, a reduction more than 50% isn't possible.

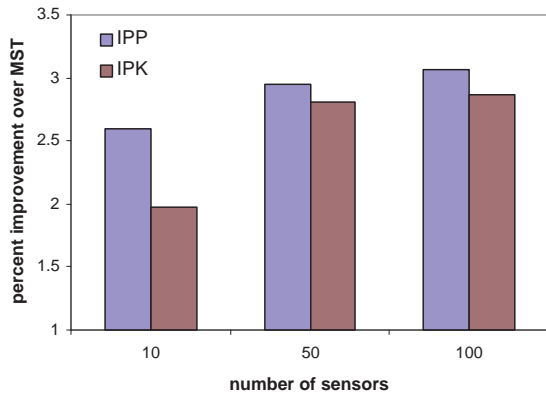
Our experimental results are consistent with those reported in [17]; single-edge switch reduces the power-cost of *MST* by between 5% and 6%. Although the best heuristics of [17] are able to improve upon *MST* by at most about 6.5% (on average), our ES2 enhancement provides an improvement that is twice as much! Also, with the ES2 enhancement, *IPK* generally did better than *MST* and *IPP*.

6 Conclusion

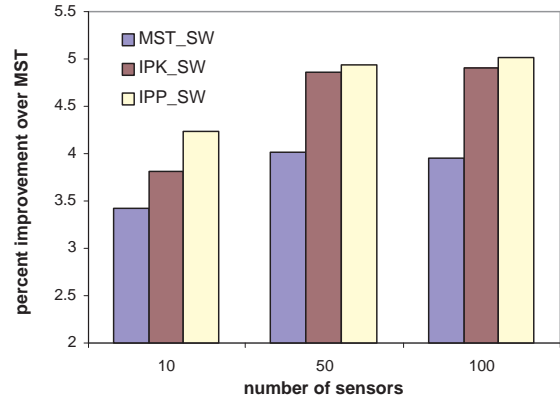
We have shown that the approximation ratio for both *IPP* and *IPK* is 2. Further, none of the enhancements—sweep, ES1a, ES1b, ES2—reduce this approximation ratio. Our experiments show that the ES2 enhancement, on average, yields a power-cost reduction (relative to *MST*) by about twice as much as obtained by the best heuristics proposed in [17].

References

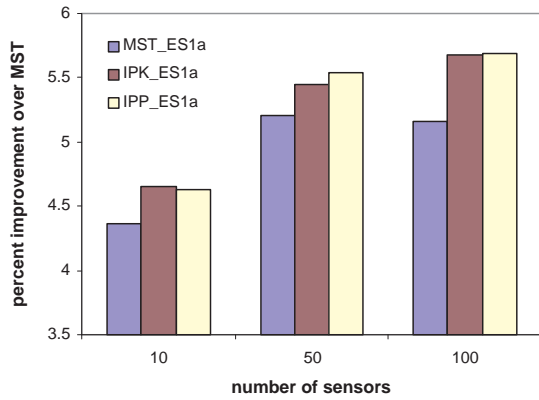
- [1] Arindam K. Das and Robert J. Marks and Mohammed El-Sharkawi, “Minimum power broadcast trees for wireless networks,” *IEEE International Symposium on Circuits and Systems*, May 2002.
- [2] O. Egecioglu and T. Gonzalez, “Minimum-energy broadcast in simple graph with limited node power,” *IASTED International Conference on Parallel and Distributed Computing and*



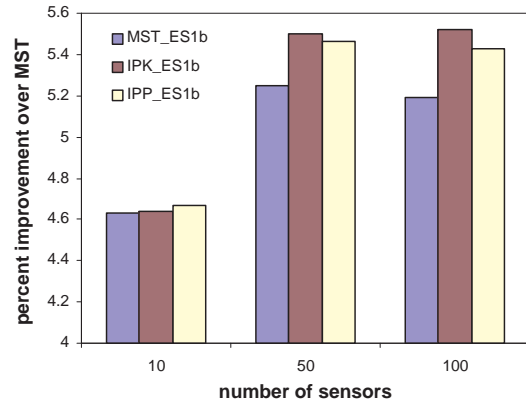
(a) Basic heuristic



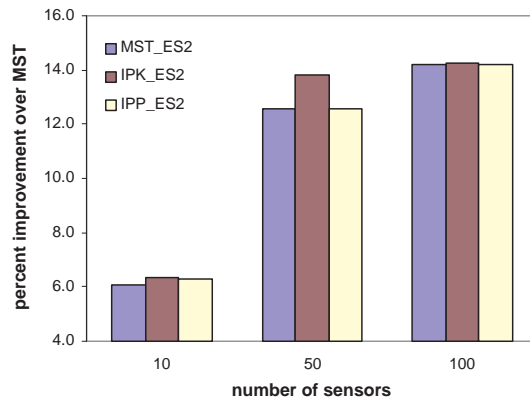
(b) With sweep



(c) With ES1a

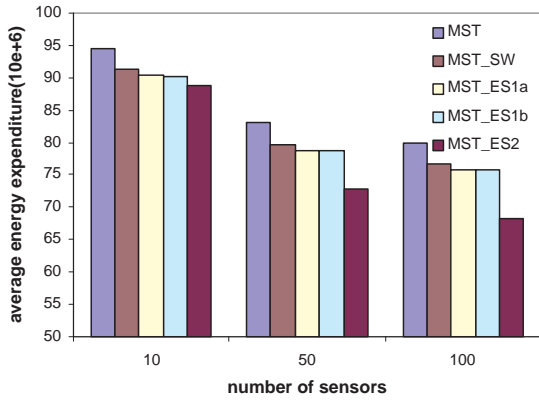


(d) With ES1b

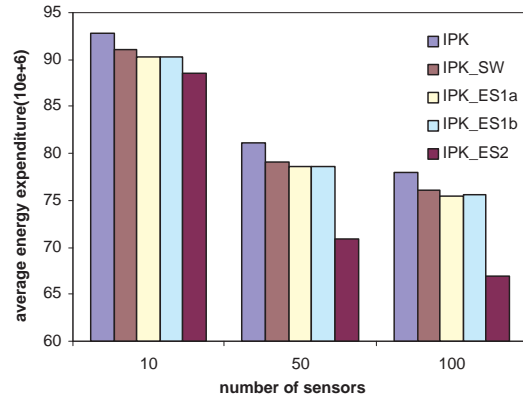


(e) With ES2

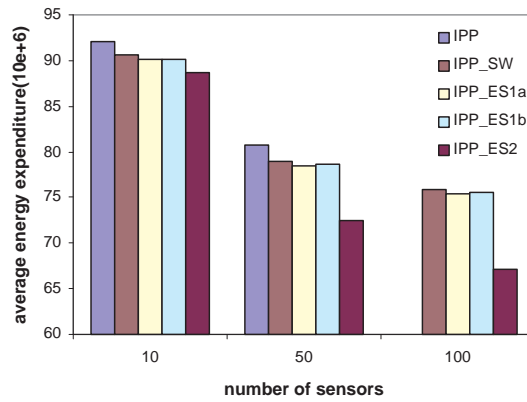
Figure 13: Average percent improvement over MST



(a) *MST*



(b) *IPK*



(c) *IPP*

Figure 14: Average power cost

Systems, vol. 30, no. 4, pp. 334–338, August 2001.

- [3] W. Chen and N. Huang, The strongly connecting problem on multihop packet radio networks, *IEEE Trans. on Comm.*, 3, 293–295, 1989.
- [4] X. Cheng, B. Narahari, R. Simha, M. Cheng, and D. Liu, Strong minimum energy topology in wireless sensor networks: NP-completeness and heuristics, *IEEE Trans. on Mobile Computing*, 2, 3, 2003, 248–256.
- [5] T. Chu and I. Nikolaidas, Energy efficient broadcast in mobile ad hoc networks, *AD-Hoc Networks and Wireless*, 2002.

	<i>MST_SW</i>	<i>MST_ES1a</i>	<i>MST_ES1b</i>	<i>MST_ES2</i>
Avg Imp(%)	4.01	5.20	5.24	12.56
Min Imp(%)	0.39	0.03	1.79	2.55
Max Imp(%)	8.26	8.26	9.35	20.78
Std Dev(%)	2.04	2.93	2.27	5.20
	<i>IPK_SW</i>	<i>IPK_ES1a</i>	<i>IPK_ES1b</i>	<i>IPK_ES2</i>
Avg Imp(%)	4.86	5.44	5.55	13.80
Min Imp(%)	-1.02	1.73	1.79	2.90
Max Imp(%)	10.06	12.32	12.42	21.63
Std Dev(%)	2.51	3.23	2.44	5.41
	<i>IPP_SW</i>	<i>IPP_ES1a</i>	<i>IPP_ES1b</i>	<i>IPP_ES2</i>
Avg Imp(%)	4.94	5.54	5.46	12.71
Min Imp(%)	-0.18	1.75	1.79	2.82
Max Imp(%)	10.57	12.16	12.16	23.27
Std Dev Imp(%)	3.36	2.47	2.39	5.60

Figure 15: Percent power-cost reduction over *MST* for $n = 50$

	<i>MST_SW</i>	<i>MST_ES1a</i>	<i>MST_ES1b</i>	<i>MST_ES2</i>
Avg Imp(%)	3.92	5.16	5.19	14.20
Min Imp(%)	0.54	2.44	1.31	3.24
Max Imp(%)	8.26	8.26	9.35	23.89
Std Dev(%)	1.32	1.63	1.83	4.43
	<i>IPK_SW</i>	<i>IPK_ES1a</i>	<i>IPK_ES1b</i>	<i>IPK_ES2</i>
Avg Imp(%)	4.86	5.68	5.51	14.23
Min Imp(%)	1.85	2.84	1.31	3.75
Max Imp(%)	7.44	10.88	10.87	24.52
Std Dev Imp(%)	1.87	1.95	1.78	4.75
	<i>IPP_SW</i>	<i>IPP_ES1a</i>	<i>IPP_ES1b</i>	<i>IPP_ES2</i>
Avg Imp(%)	5.01	5.69	5.43	14.21
Min Imp(%)	-0.10	2.14	2.77	2.38
Max Imp(%)	7.64	10.57	12.16	23.27
Std Dev Imp(%)	1.88	1.80	1.86	4.41

Figure 16: Percent power-cost reduction over *MST* for $n = 100$

- [6] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca, On the complexity of computing minimum energy consumption broadcast subgraphs. *Symp. on Theo. Aspects of Comp. Sci.*, 2001, 121–131.
- [7] A. Clementi, P. Penna, and R. Silvestri, On the power assignment problem in radio networks. *Electronic Colloquium on Computational Complexity*, 054, 2000.
- [8] M. Garey and D. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Co., 1979.
- [9] Sudipto Guha and Samir Khuller, Approximation algorithms for connected dominating sets, *Fourth Annual European Symposium on Algorithms*, 1996.
- [10] L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, Power consumption in packet radio networks. *Theo. Comp. Sci.*, 243, 289–305, 2000.
- [11] J. Park and S. Sahni, Maximum lifetime broadcasting in wireless networks, *IEEE Trans. on Computers*, 2005.
- [12] S. Sahni, Data structures, algorithms, and applications in Java, Second Edition, Silicon Press, NJ, 2005.
- [13] S. Singh and C. Raghavendra and J. Stepanek, “Power-aware broadcasting in mobile ad hoc networks,” *IEEE PIMRC’99*, Sep. 1999, Japan.
- [14] P. Wan and G. Calinescu and X. Li and O. Frieder, “Minimum energy broadcast routing in static ad hoc wireless networks,” *IEEE INFOCOM*, 2001.
- [15] J. Wieselthier and G. Nguyen and A. Ephremides, “On the construction of energy-efficient broadcasting and multicast trees in wireless networks,” *IEEE INFOCOM*, 2000.
- [16] J. Wieselthier, G. Nguyen, Algorithm for energy-efficient multicasting in static ad hoc wireless networks, *Mobile Networks and Applications*, 2001, 251-261.
- [17] E. Althaus and G. Calinescu and I.I. Mandoiu and S. Prasad and N. Tchervenski and A. Zelikovsky, “Power efficient range assignment in ad-hoc wireless networks,” *In proceedings*

of *IEEE Wireless Communications and Networking Conference (WCNC'03)*, pp. 1889–1894, 2003.

- [18] D.M. Bolugh and M. Leoncini and G. Resta and P. Santi, “On symmetric range assignment problem in wireless ad hoc networks,” *TCS 2002*, pp. 71–82, 2002.
- [19] G. Calinescu and I.I. Mandoiu and A.Z. Zelikovsky, “Symmetric connectivity with minimum power consumption in radio networks,” *TCS 2002*, pp. 119–130, 2002.
- [20] G. Calinescu and I.I. Mandoiu and A.Z. Zelikovsky, Powerpoint presentation.
“Symmetric connectivity with minimum power consumption in radio networks,” *TCS 2002*, pp. 119–130, 2002.
- [21] R. Ramanathan and R. Rosales, “Topology control of multihop wireless networks using transmit power adjustment,” *IEEE INFOCOM*, 2000.
- [22] T. Rappaport, *Wireless communications: Principles and practices*, Prentice Hall, 1996.
- [23] X. Cheng and B. Narahari and R. Simha and D. Liu, “Strong minimum energy topology in wireless sensor networks: NP-completeness and heuristics,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, 2003.