Efficient Energy Balancing Aware Multiple Base Station Deployment for WSNs

Sabbir Mahmud, Hui Wu, and Jingling Xue

School of Computer Science and Engineering
The University of New South Wales
email{sabbirm, huiw, jingling}@cse.unsw.edu.au

Abstract. Energy reduction is one of the major problems in the design of a wireless sensor network (WSN). Multiple base stations can be used to dramatically reduce the energy consumption of sensor nodes. We consider the following problem of deploying k base stations in a wireless sensor network: Given a wireless sensor network where the location of each sensor node is known, partition the whole sensor network into k disjoint clusters and place one base station for each cluster such that the maximum total energy consumption of any cluster is minimised. We propose the first heuristic for this problem. The time complexity of our heuristic is $O(kn^3)$, where n is the number of sensor nodes of the sensor network. In the special case where k is equal to 1, we propose a quadratic-time algorithm for optimally deploying the base station. Our simulation results show that our heuristic is efficient.

1 Introduction

A wireless sensor networks (WSN) consists of a set of sensors nodes that communicate with each other via radio signals. All the sensor nodes works cooperatively to monitor physical or environmental conditions, such as temperature, sound, pressure and motion. The applications of WSNs range from area monitoring, environmental monitoring, to agriculture and structural monitoring. In some applications, such as border surveillance, bushfire detection and traffic control, several thousands of sensor nodes might be deployed over the monitored region. The diameter of the monitored region can be several kilometres.

In wireless sensor networks, sensor nodes are battery powered. Most of the energy of a sensor node is consumed by communications. One key factor for the energy consumption of a sensor node is the communication distance. A sensor node consumes significantly more energy when the communication distance is increased [1]. As a result, multi hop communication between each sensor and the base station is more desirable in a large scale wireless sensor networks than the single hop communication. In multi hop communication, a sensor node may spend most of its energy on relaying data packets. Hence, it is important to shorten the hop distance between each source sensor node and the base station. The hop distances can be greatly reduced by deploying multiple base stations. All the sensor nodes are partitioned into multiple disjoint clusters with one base station for each cluster. Each sensor node sends its data only to its designated base station. Moreover, the location of the base station of each cluster is very important. If the base station is deployed far from the data sources, many sensor nodes are required to relay data packets and the energy consumption of those sensor nodes will be significantly increased. Therefore, it is an important design issue to find the best location of a base station. Nevertheless, the problem of optimally deploying multiple base stations can be reduced to the k-center problem which is NP-complete [12]. Therefore, a polynomial time algorithm is unlikely to exist.

In this paper, we study the problem of deploying k base stations such that the total energy consumption of a WSN is uniformly distributed among all the clusters. Under our energy consumption model, the total energy consumption of each cluster is a monotonically increasing function of the total shortest hop distance of all the sensor nodes of the cluster. The longer the total shortest hop distance, the more the energy consumption of a cluster. In the special case where there is only one base station, we propose a quadratic-time algorithm to optimally place a base station such that the total energy consumption of all the sensor nodes is minimised. Based on the optimal algorithm for the single base station problem, we propose a novel heuristic that aims to partition all the sensor nodes into k disjoint clusters such that the maximum total energy consumption of any cluster is minimised. We have simulated our heuristic on 195 instances of different distributions. Our simulation results show that our heuristic is very effective.

2 Definitions and Network Model

A wireless sensor network consists of a set of n identical sensor nodes each of which is located in a $2D$ plane. The location of each sensor node is known. All the sensor nodes have the same maximum communication distance R. We assume that there are no communication barriers between any two adjacent sensor nodes [1]. Therefore, a sensor node v_i can directly communicate with a sensor node v_j if the Euclidean distance between v_i and v_j is not greater than R. There are k base stations to be deployed in a target WSN. As a result, all the sensor nodes need to be partitioned into k clusters with one base station for each cluster. A sensor node in each cluster sends its data to its base station only. If the Euclidean distance between a sensor node and its base station is greater than R, the data of the sensor node must be transmitted via other sensor nodes to the base station.

Definition 1. *The connectivity graph of a wireless sensor network is a undirected graph $G =< V, E >$, where $V = \{v_i : i = 1..n$ and v_i is a sensor node$\}$, and $E = \{(v_i, v_j) :$ if the Euclidean distance between v_i and v_j is not greater than $R\}$.*

Without loss of generality, we assume that the connectivity graph G of the target wireless sensor network is connected.

Definition 2. *Given two sensor nodes v_i and v_j, the shortest hop distance from v_i to v_j is the length of the shortest path from v_i to v_j in the connectivity graph.*

The shortest hop distance of a sensor node v_i to the base station gives the lower bound on the number of hops of a packet transmitted from v_i to the base station.

[1] Our approach can be modified to handle the communication barriers.

Definition 3. *Given a cluster of sensor nodes and a base station, the total shortest hop distance of the cluster is the sum of all the shortest hop distances from each sensor node to the base station.*

Let P be a set of n distinct points called sites, in a $2D$ plane. The Voronoi diagram [11] of P is the subdivision of the plane into n cells, one for each site. A point q lies in the cell of a site $p_i \in P$ iff the Euclidean distance between q and p_i is less than the Euclidean distance between q and p_j ($p_j \in P$ and $i \neq j$). The edges of the Voronoi diagram are all the points in the plane that are equidistant to the two nearest sites.

Definition 4. *A sensor node v_i is a neighbour of a sensor node v_j if the Voronoi cells of v_i and v_j share a Voronoi edge.*

Definition 5. *Let V be a set of n sensor nodes in a $2D$ plane and $C_i(i = 1, 2, \cdots, k)$ be k disjoint clusters of V. A cluster C_i is a neighbour of a cluster C_j if there are two sensor nodes $v_s \in C_i$ and $v_t \in C_j$ such that v_s is a neighbour of v_t.*

Definition 6. *Given a cluster C_i of sensor nodes and a sensor node $v_j \notin C_i$, the Euclidean distance from v_j to C_i, denoted $d(v_j, C_i)$, is $\min\{d(v_k, v_j) : v_k \in C_i$ and $d(v_k, v_j)$ is the Euclidean distance between v_k and $v_j\}$.*

Definition 7. *Given a wireless sensor network and a point p on a $2D$ plane, the unit sensor density of p is the number of sensor nodes that are one hop away from p. The maximum unit sensor density of the wireless sensor network is the largest unit sensor density of all the points on the $2D$ plane.*

Throughout this paper, we assume that the maximum unit sensor density is a constant. In wireless sensor networks, the maximum communication distance is typically short in order to reduce the energy consumption of data transmissions. Hence this assumption is reasonable.

3 An Optimal Algorithm for Single Base Station Deployment Problem

Deploying a single base station in a cluster is a building block of our heuristic for optimally deploying k base stations. This problem is described as follows. Given a cluster of sensor nodes and a base station, find the optimal location of the base station such that the total shortest hop distance of the cluster is minimised. Next, we will propose an efficient algorithm for this problem.

The key idea of our algorithm is to find the candidate locations of the base station such that one candidate location must be the optimal location of the base station. To find all possible candidate locations, we consider each pair of sensor nodes v_i and v_j. If the Euclidean distance between v_i and v_j is greater than $2R$, where R is the maximum communication distance of all the sensor nodes, we will ignore the pair v_i and v_j. Otherwise, we find the candidate circles of v_i and v_j. A candidate circle of v_i and v_j is a circle that satisfies the following two constraints: 1) The radius of the circle is

R. 2) v_i and v_j are on its circumference. The centre of a candidate circle is a candidate location of the base station. Notice that for each pair of sensor nodes at most two candidate circles exist. If the Euclidean distance of a pair of sensor nodes is equal to $2R$, only one candidate circle of this pair exists. After finding all the candidate locations, our algorithm will search for the best candidate location of the base station. The best candidate location is the one that minimises the total shortest hop distance of all the sensor nodes to the base station placed at this candidate location. The algorithm is shown as follows.

```
Algorithm OptimalD($V$)
Input : A set $V = \{v_1, v_2, \cdots, v_m\}$ of $m$ sensor nodes in a $2D$ plane and a base station.
Output : The optimal location of the base station such that the total shortest hop distance of all
the sensor nodes to the base station at the optimal location is minimised, and the resulting total
shortest hop distance.
begin
  $C = \emptyset$;
  for each pair of sensor nodes $(v_i, v_j)(v_i, v_j \in V)$ do
      if the Euclidean distance between $v_i$ and $v_j \leq 2R$ then
          Find the candidate circles $C_1$ and $C_2$ of $v_i$ and $v_j$;
          Let $c_1$ and $c_2$ be the centres of $C_1$ and $C_2$;
          $C = C \cup \{c_1\} \cup \{c_2\}$;
  for each candidate location $c_i \in C$ do
      Place the base station at $c_i$;
      Construct the connectivity graph $G(V \cup \{c_i\})$ of all the sensor nodes and the base station;
      Compute the total shortest hop distance TSHD($c_i$) of all the sensor nodes in $V$
      to the base station located at $c_i$;
  Let $c_j$ be the candidate location with the minimum total shortest hop distance;
  return ($c_j$,TSHD($c_j$));
end
```

Theorem 1. *Given a cluster of m sensor nodes, the time complexity of the algorithm OptimalD(V) is $O(m^2)$.*

Proof. For a cluster with m sensor nodes there are $m(m-1)/2$ pairs of sensor nodes. Therefore, it takes $O(m^2)$ time to find all the candidate locations. At most two candidate locations exist for each pair of sensor nodes. Under our assumption on the maximum unit sensor density, for each sensor node v_i the number of neighbouring sensor nodes which are two hops away from v_i is at most p^2, where p, the maximum unit sensor density, is a constant. Therefore, the total number of candidate locations is $O(m)$. The connectivity graph $G(V \cup \{c_i\})$ of the all the sensor nodes and the base station at the candidate location c_i can be constructed as follows. Firstly, construct the connectivity graph $G(V)$ of all the sensor nodes, which takes $O(m^2)$ time. Secondly, add a new node for the base station at the candidate location c_i and the new edges between the base station and all the sensor nodes that can directly communicate with the base station to $G(V)$. After constructing the connectivity graph $G(V \cup \{c_i\})$, we can use breadth-first search to compute the total shortest hop distance of all the sensor nodes to the base station in $O(e)$ time, where e is the number of edges in $G(V \cup \{c_i\})$. Given the maximum unit sensor density p, $e \leq pm$ holds. Since p is a constant, it takes $O(m)$

time to compute the total shortest hop distance. As a result, the time complexity of our algorithm is $O(m^2)$.

Theorem 2. *The algorithm OptimalD*(V) *is guaranteed to find the optimal location of the base station.*

Proof. Assume that the optimal location is c_{opt}. Let $S = \{v_1, v_2, \cdots, v_r\}$ be the set of sensor nodes that are one hop away from the base station at the optimal location c_{opt}. Draw a circle C_{opt} with the radius R and the centre c_{opt}. According to the definition of the maximum communication distance R, all the sensor nodes in S must be either in C_{opt} or on the circumference of C_{opt}. Next, we show that there is a candidate location c_k generated by our algorithm such that the set of sensor nodes that are one hop away from c_k is equal to S. Consider the following three possible cases.

1. There are two sensor nodes $v_i, v_j \in S$ such that v_i and v_j are on the circumference of C_{opt}. In this case, c_{opt} is one of our candidate locations.
2. Only one sensor node $v_i \in S$ is on the circumference of C_{opt}. Turn the circle C_{opt} clockwise around v_i until another sensor $v_j \in S$ is on the circumference of C_{opt}. Now all the sensor nodes in S are still in C_{opt} and this case reduces to Case 1.
3. No sensor node is on the circumference of C_{opt}. Arbitrarily select a sensor node v_t, and move C_{opt} along the straight line $c_{opt}v_t$ until one sensor node in S is on the circumference of C_{opt}. Now all the sensors in S are still in C_{opt} or on the circumference of C_{opt}. Hence, this case reduces to Case 2.

Based on the above discussions, we can conclude that such a candidate location c_k exists. For each sensor node v_i, any path from v_i to c_k or c_{opt} must include a sensor node in S. Therefore, the shortest hop distance from v_i to c_k is equal to that from v_i to c_{opt}. As a result, c_k is also an optimal location of the base station.

4 Incremental Algorithms for Single Base Station Deployment Problems

Our heuristic for k base station deployment problem needs to repeatedly find the optimal location of a base station for a growing or shrinking cluster. A growing cluster is a cluster of sensor nodes such that a new sensor node is added to it at a time. A shrinking cluster is a cluster of sensor nodes such that a sensor node is removed from it at a time. There are two single base station deployment problems: the single base station deployment problem for a growing cluster and the single base station deployment problem for a shrinking cluster.

The single base station deployment problem for a growing cluster is described as follows: Given a cluster C_i of sensor nodes, a new sensor node v_k, and a base station, find the optimal location of the base station such that the total shortest hop distance from all sensor nodes in $C_i \cup \{v_k\}$ to the base station is minimised. A bruteforce approach to this problem is to use the algorithm proposed in the previous section, which takes $O(m^2)$ time, where m is the number of sensor nodes of the cluster. Next, we propose a faster incremental algorithm which takes $O(m)$ time.

Let $B(C_i)$ be the set of all candidate locations of the base station for C_i, SHD(v_i, v_j)

the shortest hop distance between v_i and v_j, and $N(c_j)$ the set of all neighbouring sensor nodes which are one hop away from a candidate location c_j. The incremental algorithm for the single base station deployment problem for a growing cluster is shown as follows.

```
Algorithm IncrementalGrowing($(C_i, v_k)$)
Input : A cluster $C_i$, the set $B(C_i)$ of all candidate locations of the base station for $C_i$,
the total shortest hop distance TSHD$(c_j)$ of each candidate location $c_j$ of $C_i$, the neighbour set
$N(c_j)$ of each candidate location $c_j$, and a new node $v_k$.
Output : The optimal location of the base station for $C_i \cup \{v_k\}$, the set $B(C_i)$ of all candidate
locations of the base station for $C_i \cup \{v_k\}$, the total shortest hop distance TSHD$(c_j)$ of each
candidate location $c_j$ of $C_i \cup \{v_k\}$, and the neighbour set $N(c_j)$ of each candidate location $c_j$
of $C_i \cup \{v_k\}$.
begin
  for each neighbouring candidate location $c_j$ of $v_k$ do
      $N(c_j) = N(c_j) \cup \{v_k\}$;
  Find the shortest hop distance SHD$(v_k, v_j)$ from $v_k$ to each sensor node $v_j \in C_i$;
  Construct the set $A$ of all the new candidate locations generated by $v_k$ and its neighbouring
  sensor nodes;
  for each new candidate location $c_j \in A$ do
      Find $N(c_j)$;
      Find the total shortest hop distance TSHD$(c_j)$ from all sensor nodes in $C_i \cup \{v_k\}$ to $c_j$;
  for each candidate location $c_j \in B(C_i)$ do
  // Compute the total shortest hop distance of each candidate location.
      SHD$(c_j, v_k) = 1 + \min\{$SHD$(v_s, v_k) : v_s \in N(c_j)\}$;
      TSHD$(c_j)$ =TSHD$(c_j)$+SHD$(c_j, v_k)$;
  $B(C_i) = B(C_i) \cup A$;
  Find the optimal location $c_o$ of the base station with the smallest total shortest hop distance;
  return ($c_o$,TSHD$(c_o)$);
end
```

Theorem 3. *The time complexity of IncrementalGrowing(C_i, v_k) is $O(m)$, where m is the number of sensor nodes in C_i.*

Proof. The time complexity of each part of IncrementalGrowing(C_i, v_k) is shown as follows:

- Adding v_k to the neighbour set $N(c_j)$ of each neighbouring candidate location c_j of v_k. It takes $O(m)$ time to find all the neighbouring candidate locations of v_k. The number of the neighbouring candidate locations of v_k is at most p, the maximum unit sensor density. So this part takes $O(mp) = O(m)$ time.
- Finding the shortest hop distance from v_k to each sensor node in C_i. We can use breadth-first search which takes $O(e) = O(mp) = O(m)$ time, where e is the number of edges of the connectivity graph of C_i.
- Constructing the set A. Each candidate location in A must be generated by v_k and a sensor node that is at most two hops away from v_k. The total number of sensor nodes that are one hop or two hops away from v_k is at most p^2. Each pair of sensor nodes generate at most two candidate locations. Therefore, the number of new candidate locations is at most $2p^2$. As a result, the total number of new candidate locations is $O(p^2) = O(1)$.

- Finding $N(c_j)$ for each new candidate location c_j. For each new candidate location c_j the number of sensor nodes that are one hop away from c_j is at most p. Therefore, it takes $O(p) = O(1)$ to find $N(c_j)$ for each new candidate location c_j.
- Computing the total shortest hop distance of each new candidate. It takes $O(m)$ time to compute the total shortest hop distance of each new candidate by using breadth-first search. As a result. this part takes $O(pm) = O(m)$ time.
- Computing the total shortest hop distance of each old candidate location of C_i. The shortest hop distance from v_k to each old candidate is equal to one plus the shortest hop distance from v_k to any neighbouring sensor nodes that are one hop away from this candidate. Since there are at most p such neighbouring sensor nodes and $O(m)$ old candidate locations, this part takes $O(mp) = O(m)$ time.

Based on the above discussion, we can conclude that the time complexity of our incremental algorithm IncrementalGrowing(C_i, v_k) is $O(m)$.

The single base station deployment problem for a shrinking cluster is described as follows: Given a cluster C_i of sensor nodes, a sensor node $v_k \in C_i$, and a base station, find the optimal location of the base station for the cluster $C_i - \{v_k\}$ such that the total shortest hop distance from all sensor nodes in $C_i - \{v_k\}$ to the base station is minimised. A fast incremental algorithm is shown as follows.

```
Algorithm IncrementalShrinking($(C_i, v_k)$)
Input : A cluster $C_i$, the set $B(C_i)$ of all candidate locations of the base station for $C_i$, the total
shortest hop distance TSHD($c_j$) of each candidate location $c_j$, $N(c_j)$, and a node $v_k \in C_i$.
Output : The optimal location of the base station for $C_i - \{v_k\}$, the set $B(C_i)$ of all candidate
locations of the base station for $C_i - \{v_k\}$, the total shortest hop distance TSHD($c_j$) and the
neighbour set $N(c_j)$ of each candidate location $c_j$ of $C_i - \{v_k\}$.
begin
  Find the shortest hop distance SHD($v_k, v_j$) from $v_k$ to each sensor node $v_j \in C_i$;
  Compute the set $A$ of all the candidate locations which are solely generated by $v_k$ and its
  neighbouring sensor nodes;
  $B(C_i) = B(C_i) - A$;
  for each neighbouring candidate $c_j$ that is one hop away from $v_k$ do
      $N(c_j) = N(c_j) - \{v_k\}$;
  $C_i = C_i - \{v_k\}$;
  for each candidate location $c_j \in B(C_i)$ do ;
      // Compute the total shortest hop distance of each candidate location.
      SHD($c_j, v_k$) $= 1 + \min\{$SHD($v_s, v_k$) $: v_s \in N(c_j)\}$;
      TSHD($c_j$) =TSHD($c_j$)−SHD($c_j, v_k$);
  find the optimal location $c_o$ of the base station with the minimum shortest hop distance;
  return ($c_o$,TSHD($c_o$));
end
```

Theorem 4. *The time complexity of IncrementalShrinking(C_i, v_k) is $O(m)$, where m is the number of sensor nodes in C_i.*

The time complexity analysis for IncrementalShrinking(C_i, v_k) is similar to that for IncrementalGrowing(C_i, v_k). It is omitted due to the space limitation.

5 A Heuristic for the Optimal k Base Station Deployment Problem

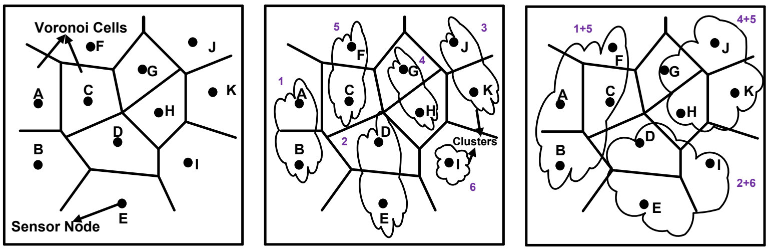
Given k base stations and a set of sensor nodes in the $2D$ plane, the energy balancing aware k base station deployment problem is to partition the whole sensor network into k disjoint clusters and deploy a base station for each cluster in an optimal way such that the maximum total shortest hop distance of any cluster is minimised. Similar to the k-center problem [12], this problem is NP-complete. Next, we will propose an efficient heuristic for this problem.

Conceptually, our heuristic works in two phases. In the first phase, it creates k initial disjoint clusters by using a greedy approach. In the second phase, it keeps moving a sensor node from a cluster with a larger total shortest hop distance to a neighbouring cluster with the smaller total shortest hop distance until a fixed point is reached.

Next, we describe how each phase works in details. In the first phase, the algorithm CreatingClusters(V, k) creates k initial disjoint clusters. It starts with creating the Voronoi diagram of all the sensor nodes. The Voronoi diagram is used to determine the nearest sensor node of a cluster. Initially, there are n clusters with one sensor node in each cluster, and the total shortest hop distance of each cluster is 0. Next, it repeatedly finds a cluster with the smallest total shortest hop distance and merges it with the best neighbouring cluster until only k clusters are left. The best neighbouring cluster is the neighbouring cluster that minimises the total shortest hop distance of the resulting cluster merged from these two clusters. The pseudo code of the algorithm is shown as follows.

```
Algorithm CreatingClusters($V, k$)
Input : A set $V = \{v_1, v_2, \cdots, v_n\}$ of sensor nodes and $k$ base stations.
Output : The k disjoint clusters and the optimal location of the base station of each cluster.
begin
  Create the Voronoi diagram for all sensor nodes in $V$;
  for each $v_i \in V$ do
      $C_i = \{v_i\}$; TSHD$(C_i) = 0$;
  NumberOfCluster$=n$;
  while NumberOfCluster $> k$ do
      Select a cluster $C_i$ with the minimum TSHD$(C_i)$.
      Fnd all the neighbouring clusters of the cluster $C_i$;
      for each neighbouring cluster $C_j$ of $C_i$ do
          $tempC_{ij} = C_i \cup C_j$;
          $(c_{ij}$, TSHD$(tempC_{ij})$ =OptimalD$(tempC_{ij})$;
      Find the neighbouring cluster $C_j$ that has the minimum TSHD($tempC_{ij}$);
      Merge $C_i$ and $C_j$ into a new cluster $C_{ij}$;
      TSHD$(C_{ij})$ =TSHD$(tempC_{ij})$;
      NumberOfCluster= NumberOfCluster$-1$;
end
```

We use an example to illustrate how the algorithm CreatingClusters(V, k) works. Consider a wireless sensor network with 11 sensor nodes and 3 base stations as shown in Figure 1. Figure 1(a) shows the Voronoi diagram our algorithm creates. All the neighbouring nodes of I are E, D, H, and K, and all the neighbouring nodes of H are C, D, G, I, J and K. At the beginning, there are 6 clusters with each sensor being one cluster. Next, the algorithm merges a smallest cluster with its best neighbouring cluster at a time. Figure 1(b) shows the intermediate clusters created by the algorithm



(a) (b) (c)

Fig. 1: An example for illustrating the algorithm CreatingClusters(V, k).

CreatingClusters(V, k), where each cluster except the cluster I is merged from two clusters. For example, the cluster $\{C, F\}$ is merged from the cluster $\{C\}$ and the cluster $\{F\}$. Figure 1(c) shows the final clusters $\{A, B, C, F\}$, $\{D, E, I\}$ and $\{G, J, H, K\}$ created by our algorithm.

In the second phase, the algorithm ClusterBalancing(C, L) aims to modify the initial clusters so that the maximum total shortest hop distance of any cluster is minimised, where C is the set of k initial clusters and L is the set of the optimal locations of the k base stations. It starts with the initial k clusters created by the algorithm CreatingClusters(V, k). In each iteration, a modifiable cluster C_i with the highest TSHD among all the clusters in C is selected. A cluster C_i is modifiable if there exist a neighbouring cluster C_s with TSHD(C_s) <TSHD(C_i) and a sensor node $v_k \in C_i$ such that TSHD($C_s \cup \{v_k\}$) <TSHD(C_i) and TSHD($C_i - \{v_k\}$) <TSHD(C_i) hold, i.e., moving v_k from C_j to C_s will reduce the maximum total shortest hop distance of both clusters. If such a modifiable cluster does not exist, all the clusters are balanced and the algorithm terminates. If such a modifiable cluster C_i exists, the algorithm will select the neighbouring cluster C_j with the smallest TSHD among all the neighbouring clusters of C_i and find the set Q of sensor nodes in C_i which are the neighbouring sensor nodes of C_j. Then it keeps moving a sensor node in Q with the smallest Euclidean distance to C_j from C_i to C_j until no sensor node in Q can be moved from C_i to C_j. A sensor node $v_k \in Q$ is moved from C_i to C_j only if v_k satisfies the following constraints:

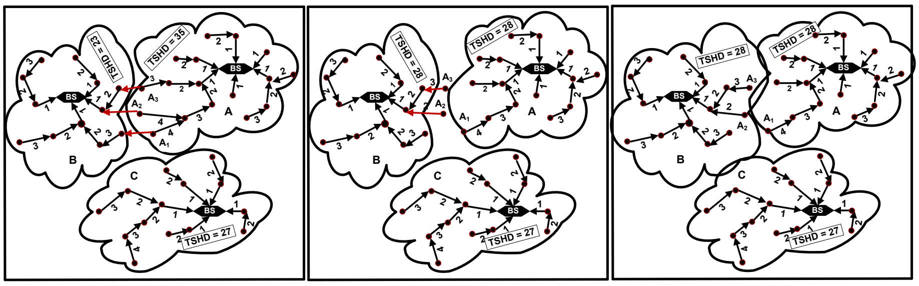
1. TSHD($C_i - \{v_k\}$)<TSHD(C_i).
2. TSHD($C_j \cup \{v_k\}$)<TSHD(C_i).

The first constraint ensures that after v_k is moved from C_i to C_j, the total shortest hop distance of C_i is reduced. If a sensor node $v_s \in Q$ is on the shortest paths of other sensor nodes in C_i to the base station, moving v_s from C_i to C_j may increase the total shortest hop distance of C_i. The second constraint guarantees that after moving v_k from C_i to C_j, the total shortest hop distance of C_j will be less than the previous total shortest hop distance of C_i. The algorithm is shown in pseudo code as follows.

Algorithm *ClusterBalancing*(C, L)
Input *: A set $C = \{C_1, C_2, \cdots, C_k\}$ of k disjoint clusters and a set $L = \{c_1, c_2, \cdots, c_k\}$ of the optimal locations of k base stations, where $c_i (i = 1, 2, \cdots, k)$ is the optimal location of the base station of the cluster C_i.*
Output *: A new set of k disjoint clusters with smaller maximum total shortest hop distance and*

```
the optimal location of the base station of each cluster.
begin
  // $A$ is the set of clusters from which no sensor node can be moved at this moment.
  // $B$ is the set of clusters from which a sensor node might be moved at this moment.
  $A = \{\}$; $B = C$; // Note that $C = A \cup B$ holds all the time.
  for each cluster $C_i \in C$ do
      $modifiable(C_i) = true$;
  while $B \neq \emptyset$ do
      Select a cluster $C_i$ with the maximum TSHD$(C_i)$ and $modifiable(C_i) = true$ from $B$;
      $S = \{C_s : C_s \in C$ and $C_s$ is a neighbouring cluster of $C_i\}$.
      Find $C_j \in S$ with the minimum TSHD($C_j$);
      $Q = \{v_s : v_s \in C_i$ and $v_s$ is a neighbouring sensor node of $C_j\}$;
      NodeMoved($C_i$) $= 0$;
      while $Q \neq \emptyset$ do
          Select a sensor node $v_s \in Q$ with the smallest Euclidean distance to $C_j$;
          if TSHD$(C_i - \{v_s\})$<TSHD$(C_i)$ && TSHD$(C_j \cup \{v_s\})$<TSHD$(C_i)$ then
              $C_j = C_j \cup \{v_s\}$; $C_i = C_i - \{v_s\}$;
              Find the new optimal locations of the base stations of $C_i$ and $C_j$;
              Recalculate TSHD$(C_i)$ and TSHD$(C_j)$;
              NodeMoved($C_i$) $= 1$;
          $Q = Q - \{v_s\}$;
      if NodeMoved$(C_i) > 0$ then
          for each neighbouring cluster $C_j$ of $C_i$ do
              if $modifiable(C_j) == false$ then
                  $modifiable(C_j) = true$; $A = A - \{C_j\}$; $B = B \cup \{C_j\}$;
      else
          $modifiable(C_i) = false$; $A = A \cup \{C_i\}$; $B = B - \{C_i\}$;
end
```

Now we use an example to illustrate how the algorithm ClusterBalancing(C, L) works. In Figure 2, there are three clusters A, B and C. The cluster A has the largest total shortest hop distance which is 35. A has two neighbouring clusters: clusters B and C. The total shortest hop distance of B is less than that of C. Therefore, the neighbouring sensor nodes from A will be moved to B. In Figure 2(a), the neighbouring sensor nodes



(a) (b) (c)

Fig. 2: An example for illustrating the algorithm ClusterBalancing(C, L).

in the cluster A are A_1, A_2, and A_3. A_3 is closest to the cluster B. TSHD($A - \{A_3\}$) is 31 which is smaller than TSHD(A). So it satisfies the first constraint. TSHD($B \cup A_3) = (23 + 3) = 26$, is smaller than TSHD($A$). So, A_3 satisfies the second constraint. As a result, A_3 is moved to the cluster B. Subsequently, the sensor node A_2 satisfies both constraints and it is also moved to the cluster B as shown in Figure 2(b). However, the sensor node A_1 does not satisfy the first condition. So it is not moved to the cluster B.

Next we analyse the time complexities of the two algorithms used by our heuristic.

Theorem 5. *The time complexity of CreatingClusters*(V, k) *is* $O(n^2 \log n)$*, where* n *is the number of sensor nodes of the wireless sensor network.*

Proof. Given n sensor nodes, its Voronoi diagram can be constructed in $O(n \log n)$ time [11]. The number of neighbouring cluster of each cluster is at most p, where p is the maximum unit sensor density. Under our assumption, p is a constant. Therefore, it takes $O(1)$ time to find all the neighbouring clusters of each cluster. Each merge takes $O(s^2)$ time if the number of sensor nodes of the resulting cluster is s. The whole merge process of clusters can be represented by a merge tree, where each node denotes merging two clusters into one cluster. At each level of the merge tree, the total work is $O(n^2)$. Since the merge tree is a balanced tree, its height is at most $\log n$. Therefore, the total work of the whole merge process is $O(n^2 \log n)$. As a result, the time complexity of the algorithm CreatingClusters(V, k) is $O(n^2 \log n)$.

Theorem 6. *The time complexity of ClusterBalancing*(C, L) *is* $O(kn^3)$*, where* n *is the number of sensor nodes of the wireless sensor network.*

Proof. It takes at most $k - 1$ sensor motions to reduce the number of sensor nodes of the cluster with the maximum total shortest hop distance by one. Therefore, the total number of sensor node motions is bounded by $O(kn)$. For each sensor motion, it takes $O(n^2)$ time to find the sensor node v_s of C_i that has the shortest Euclidean distance to C_j by using exhaustive search, and $O(n)$ time to move v_s from C_i to its neighbouring cluster C_j by using our incremental algorithms IncrementalGrowing(C_i, v_k) and IncrementalShrinking(C_i, v_k). Therefore, the time complexity of ClusterBalancing(C, L) is $O(kn^3)$.

6 Related Work

Deploying multiple base stations in a large scale senor network can significantly decrease the energy consumption of the sensor nodes by shortening the distance between the source sensor nodes and the base station. The problems of finding the best locations of multiple base stations have been studied in a number of papers under different optimisation objectives. Most papers formulate the problems as an integer linear programming (ILP) problem [6, 2, 3]. [6] proposes a heuristic for deploying multiple mobile base stations to maximise the lifetime of the sensor network. The total lifetime of the network is divided into equal period of time known as rounds and all mobile base stations change their locations at the beginning of every round. An ILP formulation is proposed to find the locations of base stations such that the maximum energy spent by each node in a round is minimised. [2] proposes a heuristic for maximising the life time of a WSN. The heuristic consists of a LP formulation for positioning multiple base stations in a sensor network and an ILP formulation for routing traffic flow from all of the sensors to these multiple sink nodes. Since the ILP problem is NP-complete, the ILP-based approaches are not applicable to large scale WSNs.

[3] proposes two-tier WSNs where the entire network is divided into clusters and each cluster has its own cluster head which is responsible for transferring data to the base station after collecting data from the sensor nodes. An iterative algorithm SPINDS is proposed to iteratively move the cluster head to a better location in order to increase the life time of the WSN. [7] studies the problems of hybrid sensor networks with resource-rich (micro-servers) and resource-deprived sensor nodes. An iterative tabu-search based algorithm is proposed to find the best locations of the micro-servers.

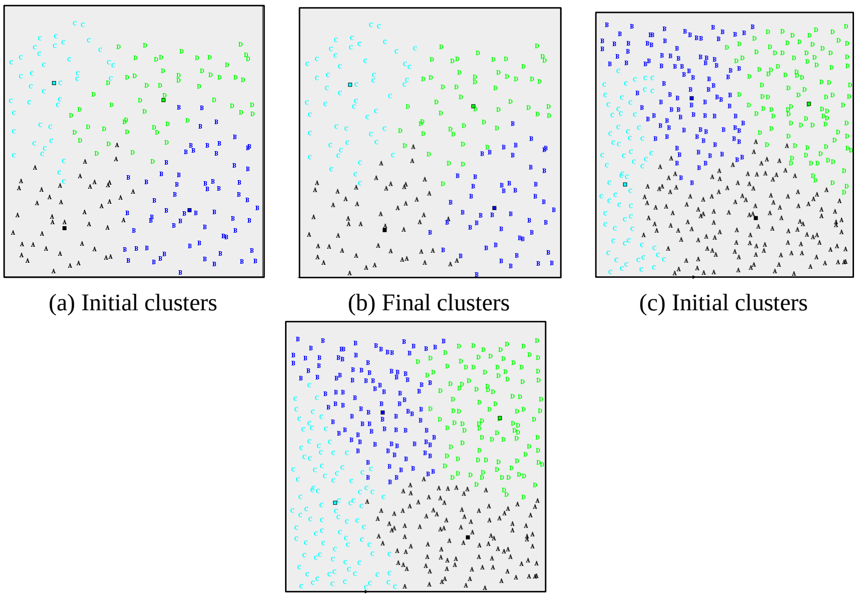
[10] propose an algorithm and a heuristic for placing k base stations in an optimal way such that the average Euclidean distance between the sensor nodes and their base stations is minimised. The algorithm assumes that each base station knows the locations of all sensor nodes and the heuristic assumes that each base station only knows the locations of its neighbouring sensor nodes and other base stations. In WSNs, it is possible for a sensor node with a shorter Euclidean distance to its base station to have a longer hop distance to its base station. Even worse, it is possible that no sensor node can communicate with the base station at the location that minimises the average Euclidean distance of all the sensor nodes to the base station. Consider a WSN with a ring topology, i.e., all the senor nodes are located on a ring. If the radius of the ring is greater than the maximum communication distance of the sensor nodes, no sensor nodes can communicate with the base station at the center of the ring. As a result, it is not feasible to minimise the average Euclidean distance between the sensor nodes and their base stations in order to minimise the lifetime or the total energy consumption of a WSN.

[9] studies the problem of placing k base stations in an optimal way such that the total latency of all the sensor nodes to their gateways is minimised. The authors proposed two heuristics for the problem using genetic algorithms. The problem with minimising the total latency of all the sensor nodes to their gateways is that it may result in unbalanced energy consumption of all the clusters.

In a WSN with multiple base stations, base stations should be deployed such that the total energy consumption of the whole WSN is distributed uniformly among all the clusters in order to increase the life time of the WSN. To our knowledge, no previous research on deploying multiple base stations with such an optimisation objective has been reported. Our work presented in this paper is the first one on uniformly distributing the total energy consumption among all the clusters.

7 Simulation Results

In this section, we evaluate the performance of our heuristic via simulations. We have generated 195 different network instances. The WSN represented by each instances is connected. All these instances are classified into three categories: grid, uniform distribution and random distribution. We have used three different numbers of base stations, i.e., 2, 4 and 6. For uniform and random distributions, we have varied the number of sensor nodes from 100 to 600 with an increment of 20 sensor nodes. For either distribution, we have generated 25 instances with 3 different numbers of base stations. For grid, we have generated 45 instances.



(a) Initial clusters

(b) Final clusters

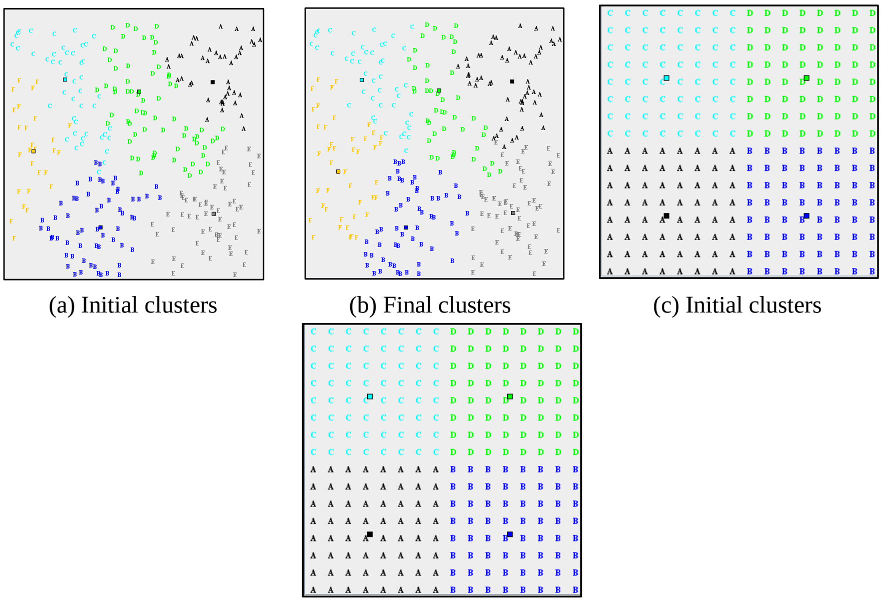
(c) Initial clusters

(d) Final clusters

Fig. 3: Simulation results for 200 and 400 sensor nodes with 4 base stations in uniform distribution.

In order to simulate our heuristic, we have used a computer with Intel Core 2 Duo processor. The processor has a clock frequency of 3 GHz and 4 GB RAM. In order to measure the performance of our heuristic, we have introduced a metric named *unbalance factor*. Given a WSN N with k clusters, the unbalance factor of N is defined as (TSHD$_{max}$−TSHD$_{min}$) / TSHD$_{max}$, where TSHD$_{max}$ $= \max\{\text{TSHD}(C_i) : C_i \in N\}$ and TSHD$_{min}$ $= \min\{\text{TSHD}(C_i) : C_i \in N\}$. The unbalance factor shows how unbalanced the k clusters of a WSN are. The smaller the unbalance factor, the more balanced the k clusters. If the unbalance factor is 0, the k clusters are fully balanced. We have also recorded the running time of our heuristic for each instance we have generated.

In order to give readers some intuition on the performance of our heuristic, we have randomly selected 4 instances among all 195 instances we have generated. The simulations results of these 4 instances are shown in Figure 3 and 4, where the sensor nodes in the same cluster are shown in the same letter and colour, and a square denotes a base station. Figure 3(a) shows the initial clusters of 200 sensor nodes with 4 base stations, generated by our algorithm CreatingClusters(V, k). Figure 3(b) shows the final clusters generated by our algorithm ClusterBalancing(C, L), where all the clusters are almost



(a) Initial clusters

(b) Final clusters

(c) Initial clusters

(d) Final clusters

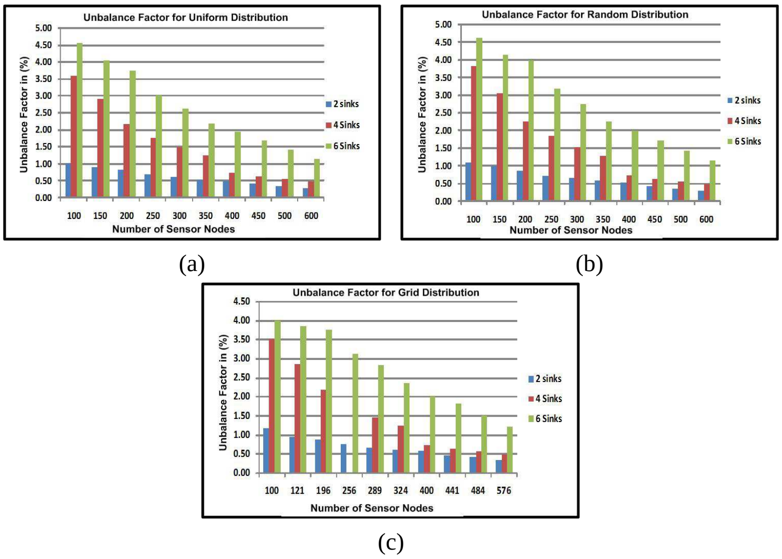
Fig. 4: Simulation results for 300 and 256 sensor nodes with 6 and 4 base stations in random and grid distributions.

balanced. Figure 3(c) and (d) show the initial clusters and the final clusters, respectively, of 400 uniformly distributed sensor nodes with 4 base stations. Figure 4(a) and (b) shows the simulation results for 300 sensor nodes with 6 base stations where sensor nodes are deployed in random distribution. Figure 4(c) and (d) show the initial clusters and the final clusters of 256 sensor nodes and 4 base stations deployed in grid, where both clusters are fully balanced.

The complete simulation results are shown in Figure 5 and 6. Figure 5 shows the unbalance factors of the clusters constructed by our heuristic for all the instances we have generated. Overall, all the clusters constructed by our heuristic are well balanced. We can observe the following patterns:

1. The unbalance factor increases with the number of base stations. The unbalance factor is at most 1% when the number of base stations is 2. The reason is that when the number of base stations increases, the relative difference between the cluster with the longest shortest hop distance and the cluster with the smallest shortest hop distance will increase.
2. Given a fixed number of base stations, the unbalance factor decreases with the number of sensor nodes. This is because the maximum total shortest hop distance of any cluster increases with the number of sensor nodes.

Figure 6 shows the running times in second of our heuristic for different instances. It shows that running time increases approximately cubically with the number of sensor



(a) (b)

(c)

Fig. 5: Unbalance Factor for sensor nodes in uniform, random and grid distributions.

nodes in all the three distributions, which is consistent with the time complexity of our heuristic. For the instance that has 600 sensor nodes in random distribution and 6 base stations, our heuristic took around 38 seconds, which is the longest running time.

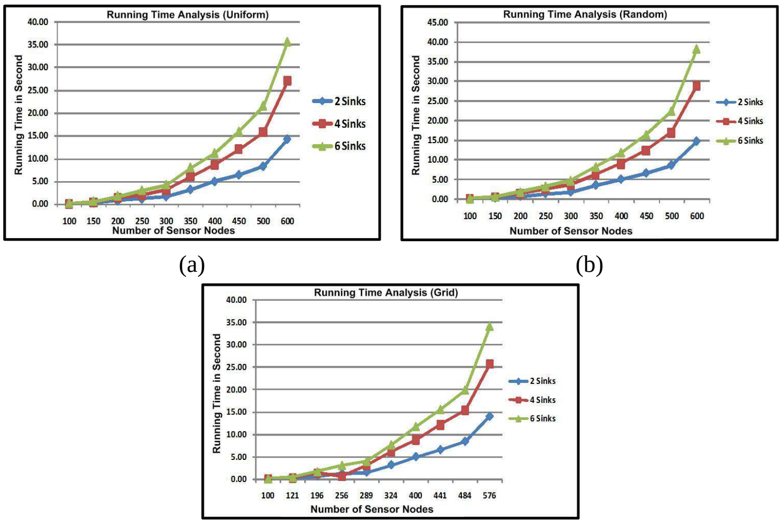
8 Conclusion

In this paper, we proposed the first heuristic for optimally deploying k base stations in a WSN such that the maximum total shortest hop distance of any cluster is minimised. The time complexity of our heuristic is $O(kn^3)$, where n is the number of sensor nodes of the WSN. In the spacial case where there is only one base station, we proposed an optimal algorithm for this problem. We have performed simulations of our heuristic on 195 instances. The simulation results show that our heuristic is very effective.

Although our heuristic performs very well, its approximation ratio is unknown. We conjecture that it is at most 2. In the future work, we will find the approximation ratio of our heuristic. Another open problem is to optimally deploy multiple base stations in a WSN where sensor nodes have variable communication ranges.

References

1. Q. Gao, K. J. Blowa, D. J. Holdinga, I. W. Marshallb, and X. H. Penga. Radio range adjustment for energy efficient wireless sensor networks. Ad Hoc Networks, Volume 4, Issue 1, 2006, pp. 75-82.
2. Kim, H, Seok, Y, Choi, N, Choi, Y and Kwon, T. Optimal multisink positioning and energy-efficient routing in Wireless Sensor Networks. Lecture Notes in Computer Science (LNCS), vol. 3391, pp. 264-274.



(a) (b)

(c)

Fig. 6: Running times for all the instances.

3. Hou, Y. T., Yi Shi, Sherali, H. D. Midkiff, S. F. On energy provisioning and relay node placement for wireless sensor networks. IEEE Transactions on Wireless Communications, volume 4, no. 5, Sept. 2005, pp. 2579-2590.
4. Liu, J, Reich, J, and Zhao, F. Collaborative in-network processing for target tracking. EURASIP Journal on Applied Signal Processing, 2003, 378–391, 2003, Hindawi Publishing Corp.
5. Alon Efrat, Sariel Har-Peled, Joseph S. B. Mitchell. Approximation Algorithms for Two Optimal Location Problems in Sensor Networks. In Proceedings of the 3rd International Conference on Broadband Communications, Networks and Systems, 2005.
6. Gandham, S, R, Dawande, M, Prakash, R, and Venkatesan, S. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In Proceedings of IEEE Global Telecommunications Conference, 2003, vol. 1, pp. 377-381.
7. Hu, W, Chou, C, T, Jha, S and Bulusu, N. Deploying long-lived and cost effective hybrid sensor networks. 1st Workshop on Broadband Advanced Sensor networks.
8. Qiu, L, Chandra, R, Jain, K and Mahdian, M. Optimizing the placement of integration points in multi-hop wireless sensor networks. In Proceedings of International Conference on Network Protocols (ICNP), 2004.
9. Yousef, W and Younis, M. Intelligent gateways placement for reduced data Latency in Wireless Sensor Networks. IEEE International Conference on Communications (ICC), 2007, pp. 3805-3810.
10. Vincze, Z, Vida, R and Vidacs, A. Deploying multiple sinks in multi-hop Wireless Sensor Networks. IEEE International Conference on Pervasive Services (ICPS), 2007, pp. 55-63.
11. J. R. Sack and J. Urrutia, Handbook of Computational Geometry, Elsevier Science, Netherlands (2000).
12. Dordit S. Hochbaum and Favid B. shmoys. A best possible heuristic for the k-center problem. Mathematics of Operations Research, Vol. 16, No. 2, May 1985, pp. 180-184.