Approximation Algorithms for Constrained Relay Node Placement in Energy Harvesting Wireless Sensor Networks

Satyajayant Misra[†], *Member, IEEE*, Nahid Ebrahimi Majd[†] and Hong Huang[‡], *Member, IEEE*

Abstract—The constrained relay node placement problem in a wireless sensor network seeks the deployment of a minimum number of relay nodes (RNs) in a set of candidate locations in the network to satisfy specific requirements, such as connectivity or survivability. In this paper, we study the constrained relay node placement problem in an energy-harvesting network in which the energy harvesting potential of the candidate locations are known *a priori*. Our aim is to place a minimum number of relay nodes, to achieve connectivity or survivability, while ensuring that the relay nodes harvest large amounts of ambient energy. We present the connectivity and survivability problems, discuss their NP-hardness, and propose polynomial time $\mathcal{O}(1)$-approximation algorithms with low approximation ratios to solve them. We validate the effectiveness of our algorithms through numerical results to show that the RNs placed by our algorithms harvest 50% more energy on average than those placed by the algorithms unaware of energy harvesting. We also develop a unified-mixed integer linear program (MILP)-based formulation to compute a lower bound of the optimal solution for minimum relay node placement and demonstrate that the results of our proposed algorithms were on average within 1.5 times of the optimal.

Index Terms—Energy harvesting, relay node placement, wireless sensor networks, constant approximation algorithm.

1 Introduction

A wireless sensor network consists of sensor nodes (SNs) and base stations (BSs). Generally, the low-cost and low-power SNs perform short range communications to conserve their limited battery resources. Given the constraints on an SN's power, the transmission of data to the BS is performed in cooperation with other SNs that forward the data. To help the power-constrained SNs, use of nodes with higher power called relay nodes (RNs) has been proposed in the literature. The placement of RNs has been used to improve properties, such as network connectivity by Cheng *et al.* [5] and Lloyd *et al.* [23], survivability by Bredin *et al.* [2], Hao *et al.* [12], and Kashyap *et al.* [16], and lifetime maximization by Cheng *et al.* [4] and Hou *et al.* [13]. These problems have been widely studied as the Relay Node Placement (RNP) problem, which seeks the placement of a minimum number of relay nodes anywhere in the deployment region to satisfy the desired objectives [11], [13].

In this paper, we consider a more practical setup – the constrained RNP. Most deployment regions in reality may contain forbidden regions or impose lower bounds on internode distances – we cannot place nodes anywhere we want. The positions of the nodes are constrained as per the restrictions in real world [24], [36]. For instance, in a volcano monitoring system, an SN/RN cannot be placed inside the crater, but may be constrained to be placed only in some allowed regions in the rim of the volcano. In this paper, we assume that the continuous unforbidden (or allowable) regions in the deployment area are discretized into *candidate locations*, thus converting the intractable continuous problem into a tractable combinatorial optimization problem. However, the number of candidate locations is unbounded; at the limit (number of candidate locations tending to infinity) they represent the allowable regions. Hence, the Constrained Relay Node Placement problem (CRNP) is concerned with placing a *minimum* number of RNs, in a subset of these candidate locations, to satisfy the desired objectives.

The advent of wireless nodes that can harvest ambient energy (solar, wind, thermal) [6], [21], [34], to supplement their battery power, has added yet another dimension to the research in wireless networks. All WSNs will possibly harvest ambient energy in the future as the added cost of the harvesting hardware/software would be much less than the cost of replacing dead batteries. Additionally, these networks can have better performance and network lifetime than non-harvesting networks [14], [15] making them attractive for additional tasks, such as multimedia and real-time communication. The existing placement algorithms [11], [13], [24], [37] cannot handle the additional complexity of energy harvesting, and as we show in Section 4, may have arbitrarily bad performance.

The RNP problems need to be re-explored in light of this new dimension. For instance, in a WSN, with nodes harvesting solar energy, a node in direct sunlight would harvest much more energy than a node in shade; thus the position of a node may have a significant bearing on its energy harvesting capabilities. This makes it imperative for placement algorithms to account for the energy harvesting potential of a node placed at different possible locations so that they may perform efficient placement that increases the energy available to the nodes and consequently increases network effectiveness. In effect, to fully-tap the potential of energy harvesting networks we need algorithms that are '*energy harvesting aware*.' These reasons **motivate** our study – **Constrained Relay Node Placement in Energy Harvesting WSNs (EH-CRNP)**. The problem is concerned with placing a minimum number of RNs in a subset of the candidate locations with high energy harvesting potential, in order to

The authors are with the New Mexico State University, Las Cruces, NM. [†] *Both authors are with the Computer Science department. Email:*{misra,nmajd}*@cs.nmsu.edu.* [‡] *Is with the Electrical Engineering department. Email:*{hhuang}*@nmsu.edu.* *This work was supported in part by the U.S. National Science Foundation under Grant 1248109. The information reported here does not reect the position or the policy of the federal government.*

achieve the objectives of connectivity or survivability. We note that EH-CRNP is more complex than the classical RNP problem (NP-hard in itself), as in addition to the number of RNs being the minimum, their placement needs to be in locations that allow them to harvest more energy.

Based on the network topology, RNP problems may be categorized into a 1-tiered or a 2-tiered placement. In a 1-tiered topology, the SNs and RNs can forward packets received from other nodes [23]. In a 2-tiered topology, only the RNs forward packets, the SNs transmit only their own packets [37]. We study the 1-tiered topology only. In a 1-tiered WSN, EH-CRNP for *connectivity* ensures that the RNs are placed in the network to connect the SNs and BSs; whereas, the placement for *survivability* ensures that the network of SNs and BSs remain connected despite a single node failure (biconnected network [7]). In both cases, the RNs' positions are chosen such that their combined energy harvesting potential is high. We define $r > 0$ and R and as the communication ranges of SNs and RNs respectively. For wider application, we study the general version of the placement problem, where $R \geq r$ [23].

Our contributions can be summarized as: (i) *We study the complexity of the EH-CRNP problem, for connectivity and survivability respectively, in a* 1*-tiered topology.* **(ii)** *We design efficient energy harvesting-aware algorithmic frameworks, and use them to propose polynomial time $\mathcal{O}(1)$-approximation algorithms (polynomial in run time and constant in approximation ratio)* to solve the problems. **(iii)** *We obtain an efficiently computable lower bound using mixed integer linear programming.* **(iv)** *Our numerical results demonstrate that our approximation algorithms perform within twice the optimal in the average case.* Our algorithms deploy a small number of RNs in a subset of candidate locations so that the overall energy harvesting potential of the deployed RNs is high and the network is connected/survivable. A more desirable objective would be to deploy a minimum number of RNs, such that the sum of the energy harvesting potential of the placed RNs is maximized. However, those type of problems are even harder to solve, and may admit only heuristic solutions, not elegant polynomial time approximation solutions. Incidentally, our proposed algorithms could serve as very good heuristics for these maximization problems as we demonstrate in our comparative study (Section 7).

Using our proposed frameworks, we present a **12.4**-approximation algorithm for connectivity and a **20**-approximation algorithm for survivability respectively. We compare our algorithms to the state of the art [24] and also to an efficiently computable mixed integer linear programming-based lower bound to the optimal number of RNs. The numerical results demonstrate that the results from our algorithms are very close to the optimal (within twice) on average in terms of the number of RNs that need to be placed, and our algorithms perform 50% better on average than the state of the art in placing the RNs in positions that allow them to harvest large amounts of energy.

In Section 2, we present the related work. Section 3 contains basic definitions and notations used in the paper. Section 4 elaborates on EH-CRNP for connectivity, while Section 5 elaborates on EH-CRNP for survivability. Section 6 presents the efficiently computable lower bound, Section 7 the comparative numerical results and analyses, and Section 8 our conclusions.

2 Related Work

In this section, we briefly describe research in RNP and energy harvesting in WSNs. We use $r > 0$ and R to denote SNs' and RNs' communication ranges respectively; and $k = 1$, $k = 2$, and $k \geq 2$ to denote network connectivity (at least one path exists between any pair of nodes) , survivability (there are at least two node-disjoint paths between any pair of nodes) , and higher order survivability (at least k node-disjoint paths), respectively. First, we present 1-tiered RNP. Lin and Xue [20] studied the RNP problem with $R = r$ and $k = 1$, and presented the Steinerization technique to solve the problem with a 5-approximation algorithm. Their work has been extended by [2], [3], [5], [11], [12], [16], [22], [23], [33], and [37]. Chen *et al.* [3] presented a 3-approximation algorithm for the problem studied in [20]. A randomized 2.5-approximation algorithm was presented by Cheng *et al.* [5]. For $R = r$ and $k = 2$, Kashyap *et al.* [16] presented a 10-approximation algorithm. Bredin *et al.* [2] proposed polynomial time $\mathcal{O}(1)$-approximation algorithms for any fixed k, $R = r$ and $k \geq 2$. Lloyd and Xue [23] were the first to study the case of $R \geq r$, for $k = 1$, prove its NP-hardness, and propose a 7-approximation algorithm. For $R \geq r$ and $k = 2$, Zhang *et al.* [37] presented a 14-approximation algorithm. The study of 2-tiered RNP was motivated by research on clustered WSNs [10], [26]. Due to lack of space, we do not discuss them further and point readers to [11], [22], [12], [23], [33].

The more practical problem of constrained relay node placement has been studied recently [24], [36]. For 1-tiered network, Misra *et al.* [24] proposed a 6.2-approximation algorithm for connectivity and a 10-approximation algorithm for survivability. Yang *et al.* [36] studied the 2-tiered problem and presented $\mathcal{O}(1)$-approximation algorithms for connectivity and an $\mathcal{O}(\log n)$-approximation algorithm for survivability. However, in our knowledge, there is no study on RNP problem in an energy harvesting WSN in the literature.

Today's SNs often have the capability to harvest ambient energy from several modes [14], [15], [21], [31]. For such networks, attention has been focused on the design of energy harvesting aware algorithms to create perpetual-operation [6], [8], [14]. Research has also been focused on energy harvesting aware routing [1]. We consider the problem of energy harvesting aware RNP, namely EH-CRNP, which to our best knowledge has not been studied in the literature. Algorithms proposed in previous studies only aim to place a minimum number of RNs and do not account for the energy harvesting capabilities of the placed nodes, hence they may perform poorly by placing the RNs at positions with low energy harvesting potentials. In this paper, we use the term non-harvesting aware (*NHA*) to refer to such algorithms. We address this gap in the literature by proposing two RNP frameworks (connectivity and survivability) that account for the energy harvesting potential of RNs placed at different locations.

3 Basic Definitions and Notations

We consider a hybrid wireless sensor network consisting of SNs, RNs, and BSs. The communication range of the SNs is $r > 0$, that of the RNs is $R \geq r > 0$, and that of the BSs is much greater than R, with the BSs in the network connected to each other. This is a valid assumption as BSs are generally connected in a WSN by wired links, the Internet, or satellites. For two nodes x and y, the Euclidean distance between x and y is denoted as $d(x,y)$, where x and y denote the locations of the nodes as well. The communication between two nodes in the graph is symmetric, hence two nodes u and v can communicate with each other if and only if $d(u, v)$ is less than or equal to the minimum of their communication ranges. Hence, an SN u is connected to an SN, RN, or a BS v if and only if $d(u, v) \leq r$. An RN u is connected to an RN or a BS v if and only if $d(u, v) \leq R$. Finally, any two BSs are always connected.

Before going further, we would like to answer a pertinent question in this context, "How do we calculate the energy harvesting potential of an RN in a candidate location?" For this information, we can take advantage of the densely deployed energy harvesting SNs to obtain an 'energy-harvesting potential map' containing all the candidate locations. The SNs in the network can communicate their energy harvesting profiles (over some duration) to a BS, if they can reach one. If the network is initially disconnected, a set-up similar to data mules [32] can be implemented to have a mobile data-gatherer travel the network and gather the energy profiles from the disconnected SNs. These profiles can be used by the BSs to estimate (mean, median, or weighted average) the energy harvesting potential (e.g., average of the instantaneous potentials from the estimate) of the candidate locations. The caveat is that when the network is disconnected (or sparse) to start with, this estimation may involve some extrapolations. We note that to reduce errors specialized estimation sensors can be strategically placed in the network. However, we do not discuss this further as we believe this problem of accurate estimation is an application-specific research problem in itself and is out of scope for this paper. We assume that we have accurately estimated the energy harvesting potential of each candidate location.

Let $\mathcal{X}$ be a set of SNs, $\mathcal{B}$ be a set of BSs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. Each node is assigned a weight that is related to its energy harvesting potential. Since we are interested in energy aware placement of RNs, for $x \in \mathcal{X} \cup \mathcal{B}$ the energy harvesting potential $e_x = 0$ and the corresponding weight $c(x) = 0$. For $z \in \mathcal{Z}$, e_z is the amount of energy an RN placed at z can harvest ($0 \leq e_z \leq e_{max}$), where e_{max} is the maximum energy a node can harvest, a variable that is determined by the hardware used. The weight of the candidate location is given by,

$$c(z) = \frac{e_{max} - e_z}{e_{max}} + 1, \tag{3.1}$$

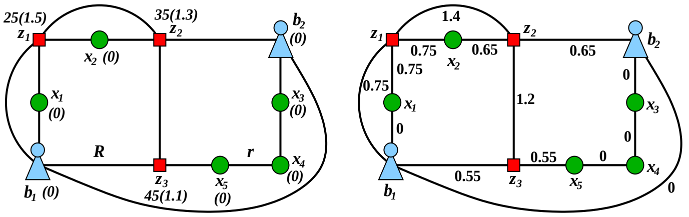
where $c(z) \in [1, 2]$. This weight function ensures that the locations that have more energy harvesting potential have lower weights, hence will be favored by our algorithms for placement. Instead of using just the fraction, we added a one to the weight function to ensure that a relay node placement comes at an additional cost (weight). Having $c(z) \in [1, 2]$ ensures that placing an RN incurs an additional cost of one per position, hence an RN will only be placed if absolutely necessary for connectivity/survivability. Our aim is to place the minimum number of RNs, the decision of whether a position is good for harvesting energy or not is the next decision. Also, having $c(z) \in [1, 2]$ helps prove the approximation bounds efficiently (refer Sections 4 and 5).

Based on the above conditions, the SNs, RNs, and BSs, with their communication ranges and their energy harvesting potentials, induce the hybrid communication graph (HCG) formally defined below.

Definition 3.1. The *edge weighted hybrid communication graph*, denoted by $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$, induced by the 6-tuple $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ is an edge weighted undirected graph with the vertex set $V = \mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}$ and the edge set E consisting of edges that satisfy the following conditions: For an SN $x \in \mathcal{X}$ and a node $y \in \mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}$, edge $(x, y) \in E$ iff $d(x, y) \leq r$. For an RN $y \in \mathcal{Z}$ and a node $z \in \mathcal{B} \cup \mathcal{Z}$, edge $(y, z) \in E$ iff $d(y, z) \leq R$. For every pair of $x, y \in \mathcal{B}$, $(x, y) \in E$. The weight of an edge (u, v) is given as

$$w(u, v) = \frac{c(u) + c(v)}{2}. \tag{3.2}$$

Hence, if $u, v \in \mathcal{X} \cup \mathcal{B}$, $w(u, v) = 0$; if exactly one of u or v belong to $\mathcal{Z}$, $0.5 \leq w(u, v) \leq 1$; and if $u, v \in \mathcal{Z}$, then $1 \leq w(u, v) \leq 2$. □



(a) HCG with node weights

(b) HCG with edge weights

Fig. 1. $HCG(\{x_1, \ldots, x_5\} \cup \{b_1, b_2\} \cup \{y_1, y_2, y_3\}, r, R, w, E)$

Each node contributes half of its weight to the weight of the edge adjacent to it. The aim is to correlate the choice of the edge to the choice of the corresponding location as will be clear when we present our algorithms.

Fig. 1(a) shows an example of a hybrid communication graph with $\mathcal{X} = \{x_1, \ldots, x_5\}, \mathcal{B} = \{b_1, b_2\}, \mathcal{Z} = \{z_1, z_2, z_3\}$. Each candidate location has an energy harvesting potential ($e_1 = 25$, $e_2 = 35$, $e_3 = 45$), with the maximum being $e_{max} = 50$. The potentials are in energy-units per second and are the average of values over some pre-determined time period. The numbers in parentheses are $c(z_1) = 1.5, c(z_2) = 1.3$, and $c(z_3) = 1.1$ and the zero weights of the SNs and the BSs. Fig. 1(b) illustrates the corresponding edge weights.

In this paper, we use $V(G)$ and $E(G)$ to denote the vertex and edge sets of a graph G.

Definition 3.2. Let $G = HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Y}, r, R, w, E)$ be an edge weighted hybrid communication graph. The relay size of G, denoted by $s(G)$, is the number of RNs in G, that is, $s(G) = |V(G) \cap \mathcal{Y}|$. For a subgraph H of G, the weight of H is defined as,

$$w(H) = \sum_{(u,v) \in E(H)} w(u, v). \tag{3.3}$$

The relay size of $s(H)$ of H is defined as,

$$s(H) = |V(H) \cap \mathcal{Y}|. \tag{3.4}$$

□

Definition 3.3. Given an undirected edge-weighted graph $G = (V, E)$, and a subset of nodes $S \subset V$, the *graph Steiner Tree Problem* (STP) seeks a minimum-weight (minimum sum of edge weights) tree that spans all nodes in S, and uses some of the nodes in $V \setminus S$. □

Definition 3.4. Given an undirected graph $G = (V, E)$ with non-negative weights on all edges $e \in E$ and a connectivity requirement $r(u, v) \in \{0, 1, 2\}$ for each pair of vertices $u, v \in V$, the $\{0, 1, 2\}-$*survivable network design problem* ({0,1,2}-SNDP) asks for a minimum weight subgraph H of G such that for any two vertices $u, v \in V$, H contains at least $r(u, v)$ vertex-disjoint paths between u and v. □

Also, a polynomial time α-approximation algorithm for a minimization problem is an algorithm $\mathcal{A}$ that, for any instance of the problem, computes a solution that is at most α times the optimal solution of the instance, in time bounded by a polynomial in the input size of the instance [7]. In this case, we also say that $\mathcal{A}$ has an approximation ratio of α. For concepts, such as NP-hard and biconnectivity please refer to [7].

4 HA-CRNP FOR CONNECTIVITY

Given a set of SNs, a set of BSs, and a set of candidate locations, we are interested in placing the *minimum number* of RNs so that the SNs and the BSs are in the same connected component of the HCG induced by the SNs, the RNs, and the BSs and the *total (sum of the) energy harvesting potential of the RNs is high*. Note again that the problem where this sum is maximized is harder to solve.

4.1 Problem Definitions and Our Contributions

Definition 4.1. Let $\mathcal{X}$ be a set of SNs, $\mathcal{B}$ be a set of BSs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. A set of RNs $\mathcal{Y} \subseteq \mathcal{Z}$ is said to be a *feasible connected maximum harvesting energy relay node placement* (denoted by **F-CME**) for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ if the $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Y}, r, R, w, E)$ is connected and $\sum_{y \in \mathcal{Y}} e_y$ is high, e_y is y's harvesting potential. The size of the corresponding F-CME is $|\mathcal{Y}|$. An F-CME is said to be a *minimum connected maximum harvesting energy relay node placement* (denoted by **M-CME**) if it has the minimum size among all F-CMEs′. □

Definition 4.2. Let $\mathcal{X}$ be a set of SNs, $\mathcal{B}$ be a set of BSs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. The connected maximum harvesting energy relay node placement problem for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$, denoted by $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$, is to find an M-CME for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$. □

We also study the special case where $\mathcal{B} = \phi$, to compare with studies that do not consider BSs in the network.

Computational Complexity: The CME problem is NP-hard. This is because, a special case of the CME problem in which each candidate location has zero energy harvesting potential, is an instance of the NP-complete critical-grid coverage problem studied by Ke *et al.* [17]. Therefore, NP-completeness of the CME problem can be proved by reduction. Consequently, we seek polynomial-time constant-approximation algorithms to solve the problem.

Contributions Summary: We present a *framework* that uses existing approximation algorithms for the graph Steiner tree problem (STP) (refer to Definition 3.3) [18], [30]. Our framework is *general and can easily incorporate* future development in algorithm design for STP. We demonstrate that using the best approximation algorithm for the STP [30], we can obtain a **12.4**-approximation algorithm for the general case and a **10.85**-approximation algorithm for the case where $\mathcal{B} = \phi$. We are the first to study the energy harvesting aware relay node placement problem and propose $\mathcal{O}(1)$-approximation algorithms for it.

4.2 An Efficient Framework of Algorithms

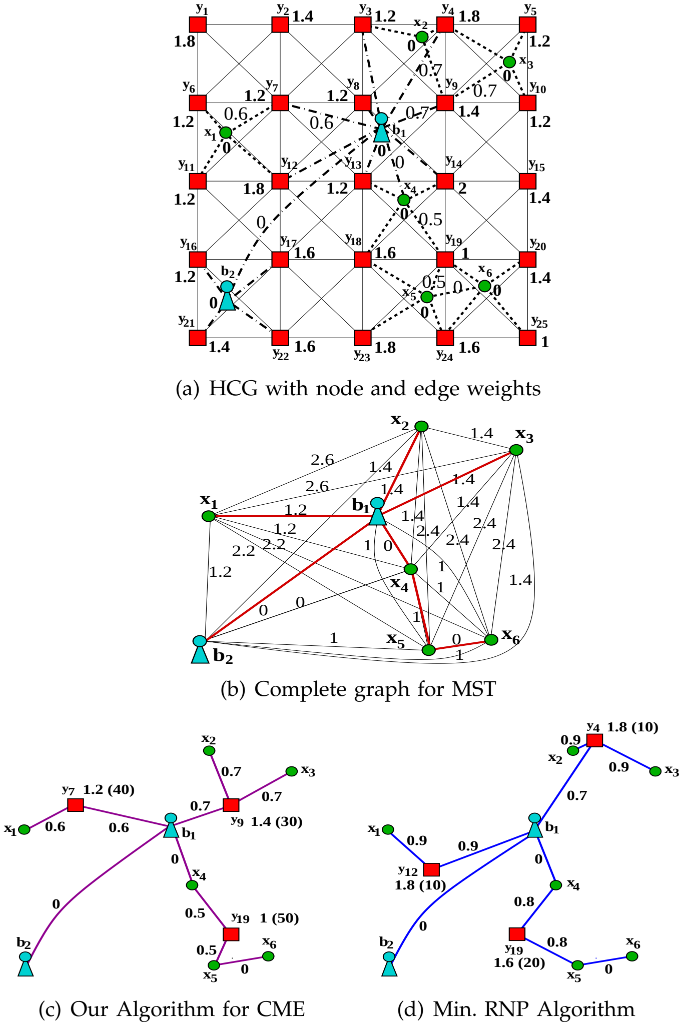
In this section, we present a polynomial time approximation algorithm for energy harvesting aware connected relay node placement. Algorithm 1 presents our approximation algorithm to solve the **CME** problem. We will show that our algorithm framework provides an 8α-approximation algorithm for the general CME problem and a 7α-approximation algorithm for the CME problem when $\mathcal{B} = \phi$, where α is the approximation ratio of an approximation algorithm $\mathcal{A}$ for the STP.

Algorithm 1 Approximation for $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$

```
Input: $R \geq r > 0$, set of BSs $\mathcal{B}$, set of SNs $\mathcal{X}$, set of RNs′
    candidate locations $\mathcal{Z}$, $e_{max}$, and an approximation
    algorithm $\mathcal{A}$ for STP.
Output: An F-CME for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ given by $\mathcal{Y}_{\mathcal{A}}$.
1: For $x \in \mathcal{X} \cup \mathcal{B}$, assign $c(x) = 0$ and for $z \in \mathcal{Z}$, assign
   $c(z) = \frac{e_{max} - e_z}{e_{max}} + 1$.
2: Construct $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$ as per Definition 3.1 and by assigning $w(u, v) = \frac{c(u)+c(v)}{2}$ for each
   edge $(u, v) \in E$.
3: if the nodes in $\mathcal{B} \cup \mathcal{X}$ are not in a single connected
   component of $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$ then
4:     The problem does not have a feasible solution.
       Stop.
5: end if
6: Apply algorithm $\mathcal{A}$ to compute a low-weight tree subgraph $\mathcal{T}_{\mathcal{A}}$ which connects all nodes in $\mathcal{B} \cup \mathcal{X}$.
7: output $\mathcal{Y}_{\mathcal{A}} = \mathcal{Z} \cap V(\mathcal{T}_{\mathcal{A}})$
```

Algorithm 1 works as follows. In Line 1, we assign the weight to each node according to Equation 3.1. In Line 2, we construct the HCG based on Definition 3.1, with the edge weights calculated as per Equation 3.2. Line 3 checks whether all the SNs and BSs belong to the same connected component. If they do not, then there is no feasible solution and the program stops. The check for connected components can be performed in linear time using the DFS algorithm [7], which makes up Lines 3–5. We apply the α-approximation algorithm $\mathcal{A}$ in Line 6 and identify the candidate locations for RNs placement in Line 7.

The best approximation algorithm for the STP problem is the 1.55-approximation (actually $1 + \frac{\ln 3}{2}$-approximation) algorithm by Robins and Zelikovsky [30]. A simpler (and easy to implement) 2-approximation algorithm is proposed in [18]. The algorithm in [18] uses a Minimum Spanning Tree (MST)-based technique to identify the Steiner points that it has to choose to obtain a connected graph. We use this algorithm for our numerical results. Given our edge and node weight functions (Equations 3.1 and 3.2), the low weight tree $\mathcal{T}_\mathcal{A}$ translates to a high energy harvesting potential for the RNs in $\mathcal{Y}_\mathcal{A}$.



(a) HCG with node and edge weights

(b) Complete graph for MST

(c) Our Algorithm for CME

(d) Min. RNP Algorithm

Fig. 2. Illustration of Algorithm 1 and comparison with results of a non-harvesting aware algorithm using an HCG with six SNs (circles), two BSs (pawn shape), and 25 candidate locations (squares).

Fig. 2 illustrates Algorithm 1. Fig. 2(a) shows an HCG, where the solid black lines represent edges between nodes in the set $\mathcal{Z}$, the dashed lines represent the edge (u, v), $u \in \mathcal{X}$ and $v \in \mathcal{X} \cup \mathcal{Z}$, and the dashed-dot lines represent the edge (b, z), $b \in \mathcal{B}$ and $z \in \mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}$. We show the node weights (bold numerals) for all the nodes, and the edge weights (normal font numerals) of only the edges that form a part of the complete graph whose edges represent the shortest path between every $u, v \in \mathcal{X} \cup \mathcal{B}$. This complete graph is obtained using the approximation algorithm in [18] as $\mathcal{A}$. The other edge weights are not shown to improve figure clarity. Fig. 2(b) shows the corresponding complete graph with the resulting MST shown using thick (red) lines. Fig. 2(c) shows the placement of the RNs by our algorithm to create $\mathcal{T}_\mathcal{A}$ (numbers in parentheses represent the harvesting potentials of the RNs in energy units/sec). Fig. 2(d) shows a possible placement of the RNs by the non-harvesting aware (NHA) CRNP algorithm [24]. Coincidentally, both the algorithms choose three RNs, however, the RNs chosen by our algorithm can harvest much more energy than RNs chosen by NHA algorithm which chooses three RNs that harvest only 10, 10, and 20 energy units/sec. *The fact that the NHA algorithms may have arbitrarily poor performance underscores the need for our energy harvesting aware algorithms.*

Theorem 4.1. The asymptotic time complexity of Algorithm 1 is given by $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2 + T(\mathcal{A}))$; $T(\mathcal{A})$ is the run time complexity of the approximation algorithm $\mathcal{A}$ for the STP.

Proof: Line 1 assigns the weights to the nodes, which can be accomplished in $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|)$ time. Line 2 constructs the HCG which requires $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2)$ time. The Depth First Search algorithm can be used on Lines 3–5, which has a complexity of $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2)$ as well. Line 6 takes $O(T(\mathcal{A}))$ time as specified, while Line 7 take $\mathcal{O}(|\mathcal{Z}|)$. Hence proved. □

To prove the approximation ratio of the algorithm, first we need to prove some necessary results. Let us define $\mathcal{T}_\mathcal{N}$ as the tree that is derived from $\mathcal{T}_\mathcal{A}$ obtained in Algorithm 1 by removing the edge weight $w(u, v)$ for each edge (u, v) in $E(\mathcal{T}_\mathcal{A})$ and by reinstating the node weight $c(u)$ for each $u \in \mathcal{X} \cup \mathcal{B} \cup \mathcal{Y}$.

Lemma 4.1. If the $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ has a feasible solution, then:
(a) $|\mathcal{Y}_\mathcal{A}| \le w(\mathcal{T}_\mathcal{N}) \le 2 \cdot |\mathcal{Y}_\mathcal{A}|$ and
(b) $w(\mathcal{T}_\mathcal{N}) \le w(\mathcal{T}_\mathcal{A})$, where $w(\mathcal{T}_\mathcal{N})$ is the node-weight sum of $\mathcal{T}_\mathcal{N}$ and $w(\mathcal{T}_\mathcal{A})$ is the edge weight sum of $\mathcal{T}_\mathcal{A}$.

Proof: (a) From Equation 3.1, we know that $1 \le c(z) \le 2$, for $z \in \mathcal{Z}$, and from Algorithm 1, $\mathcal{Y}_\mathcal{A} \subseteq \mathcal{Z}$. Hence we have,

$$|\mathcal{Y}_\mathcal{A}| \le \sum_{y \in \mathcal{Y}_\mathcal{A}} c(y) \le 2 \cdot |\mathcal{Y}_\mathcal{A}| \tag{4.1}$$

$$|\mathcal{Y}_\mathcal{A}| \le \quad w(\mathcal{T}_\mathcal{N}) \quad \le 2 \cdot |\mathcal{Y}_\mathcal{A}| \tag{4.2}$$

The weight of each RN is between 1 and 2 and the weight of all the other nodes is zero, hence the weight of the tree is bounded between $|\mathcal{Y}_\mathcal{A}|$ and $2 \cdot |\mathcal{Y}_\mathcal{A}|$.
(b) For (u, v) in $\mathcal{T}_\mathcal{A}$, $w(u, v) = \frac{c(u)+c(v)}{2}$, hence each RN gives half of its weight to each one of its edges. Since no RN $y \in \mathcal{T}_\mathcal{A}$ in the tree is a leaf, the degree of y is at least two. So y gives at least all of its weight to its edges. Additionally, the weights of SNs and BSs are 0. Therefore, $w(\mathcal{T}_\mathcal{N}) \le w(\mathcal{T}_\mathcal{A})$. □

Lemma 4.2. Let $\mathcal{T}_{opt}$ be a minimum edge weighted spanning tree (MST) of the HCG, where $\mathcal{Y}_{opt}$ is the optimal solution to the $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem, $y \in \mathcal{Y}_{opt}$ and $x \in \mathcal{X} \cup \mathcal{B}$ then,

$$\sum_{(y,x) \in E(\mathcal{T}_{opt})} w(y, x) \le 6 \cdot |\mathcal{Y}_{opt}|$$

Proof: Refer to Appendix A.1. □

Lemma 4.3. Let $\mathcal{T}_{opt}$ be an MST of the HCG, where $\mathcal{Y}_{opt}$ is the optimal solution to the $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem and $y, z \in \mathcal{Y}_{opt}$ then, $\sum_{(y,z) \in E(\mathcal{T}_{opt})} w(y, z) \le 2 \cdot (|\mathcal{Y}_{opt}| - 1)$.

Proof: Since $\mathcal{T}_{opt}$ is a tree and no RN is a leaf, there are at most $|\mathcal{Y}_{opt}| - 1$ edges with two RNs adjacent to each edge. From Equations 3.1 and 3.2, the weight of each such edge is at most two. Therefore the sum of the weight of all such edges is less than or equal to $2 \cdot (|Y_{opt}| - 1)$. □

Lemma 4.4. Let $\mathcal{T}_{opt}$ be an MST of the HCG, where $\mathcal{Y}_{opt}$ is the optimal solution to the $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem, $y, z \in \mathcal{Y}_{opt}$, and $x \in \mathcal{X} \cup \mathcal{B}$ then, $w(\mathcal{T}_{opt}) \leq 8 \cdot |\mathcal{Y}_{opt}|$.

Proof: Since only the edges incident on the RNs contribute to the weight of $\mathcal{T}_{opt}$ we have,

$$\begin{aligned} w(\mathcal{T}_{opt}) = & \sum_{(y,z)\in E(\mathcal{T}_{opt})} w(y,z) + \sum_{(y,x)\in E(\mathcal{T}_{opt})} w(y,x) && (4.3) \\ \leq & \quad 2 \cdot (|\mathcal{Y}_{opt}| - 1) + 6 \cdot |\mathcal{Y}_{opt}| && (4.4) \\ \leq & \quad 8 \cdot |\mathcal{Y}_{opt}| && (4.5) \end{aligned}$$

Inequality 4.3 implies that the weight of $\mathcal{T}_{opt}$ is a sum of the weight of the edges incident on the RNs in $\mathcal{Y}_{opt}$. Inequality 4.4 follows from Lemmas 4.2 and 4.3. □

Theorem 4.2. (a) The $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem has a feasible solution *iff* the $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$ contains $\mathcal{X} \cup \mathcal{B}$ in one connected component.
(b) If $CME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ has a feasible solution, Algorithm 1 is guaranteed to find a feasible solution, which is no more than 8α times the number of RNs in an optimal solution $\mathcal{Y}_{opt}$, where α is the approximation ratio of approximation algorithm $\mathcal{A}$ and $\mathcal{T}_{opt}$ is an MST of $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Y}_{opt}, r, R, w, E)$.

Proof: Refer to Appendix A.2. □

Theorem 4.3. The general CME problem has a 12.4-approximation algorithm, while the special case with $\mathcal{B} = \phi$, has a 10.85-approximation algorithm.

Proof: If we use the approximation algorithm proposed in [30], which has an approximation ratio of $\alpha = 1 + \frac{\ln 3}{2} \leq 1.55$ as algorithm $\mathcal{A}$, then the general case has an approximation of 8α, which is 12.4. In case where $\mathcal{B} = \phi$, it is easy to see that the approximation ratio is 7α, leading to a 10.85-approximation algorithm. □

Theorem 4.4. The general CME problem has a 16-approximation algorithm with a running time of $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2 \log(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|))$. The special case with $\mathcal{B} = \phi$, has a 14-approximation algorithm with a running time of $\mathcal{O}(|\mathcal{X} \cup \mathcal{Z}|^2 \log(|\mathcal{X} \cup \mathcal{Z}|))$.

Proof: If we use the MST-based 2-approximation algorithm for the STP proposed in [18] as $\mathcal{A}$ in Algorithm 1, then the running time of Algorithm 2 for the general case will be $\mathcal{O}(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z})^2 \log(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}))$ and the special case will be $\mathcal{O}(|\mathcal{X} \cup \mathcal{Z}|^2 \log(|\mathcal{X} \cup \mathcal{Z}|))$. The corresponding approximation ratios can be derived in the same way as in Theorem 4.3. □

5 HA-CRNP FOR SURVIVABILITY

In the last section, we discussed EH-CRNP for connectivity, in this section we will discuss the same for survivability. Given a set of SNs, a set of BSs, and a set of candidate locations where RNs can be placed, we are interested in placing the *minimum number* of RNs so that the SNs and the BSs are in the same *biconnected component* (there exist two vertex-disjoint paths between every $u, v \in \mathcal{X} \cup \mathcal{B}$) of the hybrid communication graph induced by the SNs, the RNs, and the BSs, and the *overall energy harvesting potential of the RNs is high*.

Survivable relay node placement (also known as fault tolerant relay node placement) in wireless sensor networks has been studied by many researchers [2], [12], [16]. The objective here is to ensure that the network remains connected in the event of up to one node failures. For a network to tolerate up to m node failures, it has to be $(m + 1)$-connected. Several works have considered survivability in the case where the RNP is unconstrained [2], [11], [12], [16], [22], [33], [37]. CRNP placement has only recently received attention, with connectivity ($k = 1$) and survivability ($k = 2$) having been studied in both 1-tier and 2-tier networks [24], [36]. However, to our best knowledge there has been no work on harvesting aware RNP for survivability.

5.1 Problem Definitions and Our Contributions

Definition 5.1. Let $\mathcal{X}$ be a set of SNs, $\mathcal{B}$ be a set of BSs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. A set of RNs $\mathcal{Y} \subseteq \mathcal{Z}$ is said to be a *feasible survivable maximum harvesting energy relay node placement* (denoted by **F-SME**) for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Y})$ if the graph $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$ is biconnected and $\sum_{y\in\mathcal{Y}} e_y$ is high. The size of the corresponding F-SME is $|\mathcal{Y}|$. An F-SME is said to be a *minimum survivable maximum harvesting energy relay node placement* (denoted by **M-SME**) if it has the minimum size among all F-SMEs'. □

Definition 5.2. Let $\mathcal{X}$ be a set of SNs, $\mathcal{B}$ be a set of BSs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. The connected maximum harvesting energy relay node placement problem for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$, denoted by $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$, is to find an M-SME. □

Computational Complexity: Since the corresponding connectivity problem CME is NP-hard, the SME problem, which requires biconnectivity, will be NP-hard as well (can be proved by reduction). We have also proved that the SME problem is NP-hard by reducing the planar 3-SAT [19] to the SME problem. However, due to space constraints we omit the long proof and instead concentrate on designing polynomial time approximation algorithms.

Contributions Summary: We present a *framework* that incorporates existing approximation algorithms for the {0,1,2}-Survivable Network Design Problem ({0,1,2}-SNDP) (refer to Definition 3.4) [9], [28], [29]. Our framework is *general and can easily incorporate future development* in algorithm design for {0,1,2}-SNDP. We demonstrate that using the best approximation algorithm for the SNDP [9], we can obtain a **20**-approximation algorithm for the general case and a **18**-approximation algorithm for the special case where $\mathcal{B} = \phi$.

5.2 An Efficient Framework of Algorithms

In this section, we present Algorithm 2 – a polynomial time approximation algorithm to solve the SME problem. We will show that our algorithm framework provides an 10α-approximation algorithm for the general SME problem and a 9α-approximation algorithm for the SME problem when $\mathcal{B} = \phi$, where α is the approximation ratio of an approximation algorithm $\mathcal{A}$ for the {0,1,2}-SNDP.

The algorithm is similar to Algorithm 1 in the initial setup. Line 1 assigns weight to each node as per Equation 3.1. Line 2 constructs the HCG based on Definition 3.1 and assign edge weights as per Equation 3.2. Lines 3–5 check if all SNs and BSs are in the same biconnected component. If they are not, then SME has no feasible solution. The check in

Algorithm 2 Approximation for $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$

```
Input: $R \geq r > 0$, set of BSs $\mathcal{B}$, set of SNs $\mathcal{X}$, set of RNs' candidate locations $\mathcal{Z}$, $e_{max}$, and an approximation algorithm $\mathcal{A}$ for {0,1,2}-SNDP.
Output: An F-SME for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ given by $\mathcal{Y}_{\mathcal{A}}$.
1: For $x \in \mathcal{X} \cup \mathcal{B}$, assign $c(x) = 0$ and for $z \in \mathcal{Z}$, assign $c(z) = \frac{e_{max} - e_z}{e_{max}} + 1$.
2: Construct $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$ as per Definition 3.1 and by assigning $w(u, v) = \frac{c(u)+c(v)}{2}$ for each edge $(u, v)$.
3: if the nodes in $\mathcal{B} \cup \mathcal{X}$ are not in a single biconnected component of $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$ then
4:     The problem does not have a feasible solution. Stop.
5: end if
6: Assign connectivity requirements $r(u, v)$ for each pair of vertices in $u, v \in \mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}$. For edge pair $u, v \in \mathcal{X} \cup \mathcal{B}$, $r(u, v) = 2$, for every other pair $r(u, v) = 0$.
7: Apply algorithm $\mathcal{A}$ to compute a low-weight biconnected subgraph $\mathcal{H}_{\mathcal{A}}$ which biconnects all nodes in $\mathcal{B} \cup \mathcal{X}$.
8: output $\mathcal{Y}_{\mathcal{A}} = \mathcal{Z} \cap V(\mathcal{H}_{\mathcal{A}})$
```

Line 3 can be performed by using the DFS algorithm, which can compute all biconnected components in the HCG [7]. Line 6 constructs an instance of the {0,1,2}-SNDP. Line 7 applies the α-approximation algorithm $\mathcal{A}$ to the HCG to obtain a low weight biconnected subgraph containing all SNs and BSs. In Line 8, the candidate locations are identified for RNs placement.

The best approximation algorithm for the {0,1,2}-SNDP is the 2-approximation algorithm proposed by Fleischer [9]. A simpler 3-approximation algorithm was proposed by Ravi and Williamson [28]. We use this algorithm for our numerical results.

Theorem 5.1. The asymptotic time complexity of Algorithm 2 is given by $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2 + T(\mathcal{A}))$, where $T(\mathcal{A})$ is the run time complexity of the approximation algorithm $\mathcal{A}$ for the {0,1,2}-SNDP.

Proof: Line 1 assigns the weights to the nodes, which can be accomplished in $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|)$ time. Line 2 constructs the HCG which requires $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2)$ time. The Depth First Search algorithm can be used on Lines 3–5, which has a complexity of $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2)$ as well. Line 6 has a complexity of $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2)$ for assigning the connectivity requirements. Line 7 takes $\mathcal{O}(T(\mathcal{A}))$ time as specified, while Line 8 take $\mathcal{O}(|\mathcal{Z}|)$. □

To prove the approximation ratio of the algorithm, first we need to prove some necessary results. The first two results are lemmas that we state without proving. We recommend the readers to refer to [24] for the proofs.

Lemma 5.1. Let $G(V, E)$ be an undirected biconnected graph where $|V| \geq 3$ and each edge $e \in E$ has a unit length $l(e) = 1$. Let $H(V, E')$ be a minimum length biconnected subgraph of G. Then $|E'| \leq 2|V| - 3$. □

Lemma 5.2. Let $G(V, E)$ be an undirected connected graph where $|V| \geq 3$ and each edge $e \in E$ has a unit length $l(e) = 1$, if $H(V, E')$ be a minimum length connected subgraph of G such that two vertices u and v are in the same biconnected component of H if and only if they are in the same biconnected component of G, then $|E'| \leq 2|V| - 1$. □

Let us define $\mathcal{H}_{\mathcal{N}}$ as the biconnected subgraph that is derived from $\mathcal{H}_{\mathcal{A}}$ obtained in Algorithm 2 by removing the edge weight $w(u, v)$ for each edge (u, v) in $E(\mathcal{H}_{\mathcal{A}})$ and by reinstating the node weight $c(u)$ for each $u \in \mathcal{X} \cup \mathcal{B} \cup \mathcal{Y}$.

Lemma 5.3. If the $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ has a feasible solution, then:
(a) $|\mathcal{Y}_{\mathcal{A}}| \leq w(\mathcal{H}_{\mathcal{N}}) \leq 2 \cdot |\mathcal{Y}_{\mathcal{A}}|$ and
(b) $w(\mathcal{H}_{\mathcal{N}}) \leq w(\mathcal{H}_{\mathcal{A}})$, where $w(\mathcal{H}_{\mathcal{N}})$ is the node weights sum of $\mathcal{H}_{\mathcal{N}}$ and $w(\mathcal{H}_{\mathcal{A}})$ is the edge weight sum of $\mathcal{H}_{\mathcal{A}}$.

Proof: The proof follows that of Lemma 4.1.
(a) From Equation 3.1, we know that $1 \leq c(z) \leq 2$, for $z \in \mathcal{Z}$, and from Algorithm 1, $\mathcal{Y}_{\mathcal{A}} \subseteq \mathcal{Z}$. Hence we have,

$$|\mathcal{Y}_{\mathcal{A}}| \leq \sum_{y \in \mathcal{Y}_{\mathcal{A}}} c(y) \leq 2 \cdot |\mathcal{Y}_{\mathcal{A}}| \tag{5.1}$$

$$|\mathcal{Y}_{\mathcal{A}}| \leq \quad w(\mathcal{H}_{\mathcal{N}}) \quad \leq 2 \cdot |\mathcal{Y}_{\mathcal{A}}| \tag{5.2}$$

The weight of each RN is between 1 and 2 and the weight of all the other nodes is zero, hence we have Inequality 5.2.
(b) For (u, v) in $\mathcal{H}_{\mathcal{A}}$, $w(u, v) = \frac{c(u)+c(v)}{2}$, hence each RN gives half of its weight to each one of its edges. Since each RN $y \in \mathcal{H}_{\mathcal{A}}$ belongs to the biconnected component, the degree of y is at least two. So y gives at least all of its weight to its edges. Additionally, the weights of SNs and BSs are 0. Therefore, $w(\mathcal{H}_{\mathcal{N}}) \leq w(\mathcal{H}_{\mathcal{A}})$. □

Lemma 5.4. Let $\mathcal{H}_{opt}$ be a minimum edge weighted biconnected spanning subgraph of the HCG, where $\mathcal{Y}_{opt}$ is the optimal solution to the $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem, $y \in \mathcal{Y}_{opt}$ and $x \in \mathcal{X} \cup \mathcal{B}$ then,

$$\sum_{(y,x) \in E(\mathcal{H}_{opt})} w(y, x) \leq 6 \cdot |\mathcal{Y}_{opt}|$$

Proof: Refer to Appendix B.1. □

Lemma 5.5. Let $\mathcal{H}_{opt}$ be a minimum-weight biconnected subgraph of the HCG, where $\mathcal{Y}_{opt}$ is the optimal solution to the $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem and $y, z \in \mathcal{Y}_{opt}$ then, $\sum_{(y,z) \in E(\mathcal{Y}_{opt})} w(y, z) \leq 2 \cdot (2 \cdot |\mathcal{Y}_{opt}| - 1)$.

Proof: With the help of Lemmas 5.1 and 5.2, we have that the number of edges with weight between one and two in $\mathcal{H}_{opt}$ is at most $2|\mathcal{Y}_{opt}| - 1$. These are the edges of the form (y, z), where $y, z \in \mathcal{Y}_{opt}$. Therefore, $\sum_{(y,z) \in E(\mathcal{Y}_{opt})} w(y, z) \leq 2 \cdot (2 \cdot |\mathcal{Y}_{opt}| - 1)$. □

Lemma 5.6. Let $\mathcal{H}_{opt}$ be a minimum-weight biconnected subgraph of the HCG, where $\mathcal{Y}_{opt}$ is the optimal solution to the $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem, $y, z \in \mathcal{Y}_{opt}$, and $x \in \mathcal{X} \cup \mathcal{B}$ then, $w(\mathcal{T}_{opt}) \leq 10 \cdot |\mathcal{Y}_{opt}|$.

Proof: Since only the edges adjacent on the RNs contribute to the weight of $\mathcal{H}_{opt}$ we have,

$$w(\mathcal{H}_{opt}) = \sum_{(y,z) \in E(\mathcal{H}_{opt})} w(y, z) + \sum_{(y,x) \in E(\mathcal{H}_{opt})} w(y, x) \tag{5.3}$$

$$\leq \quad 2 \cdot (2 \cdot |\mathcal{Y}_{opt}| - 1) + 6 \cdot |\mathcal{Y}_{opt}| \tag{5.4}$$

$$\leq \quad 10 \cdot |\mathcal{Y}_{opt}| \tag{5.5}$$

Inequality 5.3 implies that $w(\mathcal{H}_{opt})$ is a sum of the weight of the edges incident on the RNs in $\mathcal{Y}_{opt}$. Inequality 5.4 follows from Lemmas 5.4 and 5.5, resulting in Inequality 5.5. □

Theorem 5.2. (a) The $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem has a feasible solution *iff* the $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$ contains $\mathcal{X} \cup \mathcal{B}$ in one biconnected component.
(b) If the $SME(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ problem has a feasible solution, Algorithm 1 is guaranteed to find a feasible solution, which is no more than 10α times the number of RNs in an optimal solution $\mathcal{Y}_{opt}$, where α is the approximation ratio of the approximation algorithm $\mathcal{A}$ and $\mathcal{H}_{opt}$ is a minimum-weight biconnected subgraph of $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Y}_{opt}, r, R, w, E)$.

Proof: Refer to Appendix B.2. □

Theorem 5.3. The general SME problem has a 20-approximation algorithm, while the special case with $\mathcal{B} = \phi$, has an 18-approximation algorithm.

Proof: If we use the 2-approximation algorithm proposed in [9] as algorithm $\mathcal{A}$, then the general case has an approximation of 10α, which results in a 20-approximation algorithm. When $B = \phi$ the approximation ratio become 9α, leading to an 18-approximation algorithm. □

Theorem 5.4. The general SME problem has a 30-approximation algorithm with a running time of $\mathcal{O}(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^3 + |E| \cdot |\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}| \cdot \alpha(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|))$, where E is the edge set of the $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$. The special case with $\mathcal{B} = \phi$, has a 27-approximation algorithm with a running time of $\mathcal{O}(|\mathcal{X} \cup \mathcal{Z}|^3 + |E| \cdot |\mathcal{X} \cup \mathcal{Z}| \cdot \alpha(|\mathcal{X} \cup \mathcal{Z}|))$.

Proof: If we use the 3-approximation algorithm for the {0, 1, 2}-SNDP proposed by Ravi and Williamson [28], [29] as $\mathcal{A}$ in Algorithm 2, then the running time of Algorithm 2 follows. □

6 Efficiently Computable Lower Bound

In order to evaluate the performance of our approximation algorithms, it would be useful to compare their results with the optimal results. The lack of efficient algorithms for computing optimal solutions for **CME** and **SME** problems presents a challenge for such a comparative study. Since **CME** and **SME** are NP-hard, solving them optimally would require solving integer linear programs (ILP). Computation of optimal solutions to these ILPs in a reasonable amount of time is unlikely, unless the input size of the instances is very small. As a work around we develop a two step unified-mixed integer linear programming (unified-MILP) formulation for the problems, *which can be efficiently computed to obtain a lower bound on the number of RNs placed by the optimal solution* for the the CME and the SME problems, but in much less time than the corresponding integer programming formulations. We denote the unified-MILP formulation by MILP(**f**), where $\mathbf{f} = 1$ corresponds to the CME problem (connectivity) and $\mathbf{f} = 2$ corresponds to the SME problem (survivability). In the rest of this paper, we use the terms unified-MILP or MILP interchangeably.

We note here that the MILP provides a **lower bound** to the optimal solution (for the CME/SME problems) in terms of the minimum number of RNs *only*, it is difficult to design a formulation that also provides a lower bound to the total energy harvesting potential of the optimal solution. We will discuss why, in more detail, in this section.

TABLE 1
Notations/Variables Used In The MILP Formulation

$\mathcal{B}$	:	the set of BSs
$\mathcal{X}$	:	the set of SNs
$\mathcal{Z}$	:	the set of candidate locations for RNs
HCG	:	the hybrid communication graph $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$
$\mathbf{f}$	:	$\mathbf{f} = 1$ for connectivity and $\mathbf{f} = 2$ for survivability
t	:	BS chosen as the common sink, $t = b_1$
$\mathcal{B}'$	:	the set of BSs excluding $t = b_1$
f_{uvx}	:	variable denoting *type-x* flow on edge $(u, v) \in$ HCG
r_{zx}	:	variable denoting total flow of *type-x* ($x \in \mathcal{X}$) into $z \in \mathcal{Z}$
r_{bx}	:	variable denoting total flow of *type-x* ($x \in \mathcal{X}$) into $b \in \mathcal{B}'$
r_{wx}	:	variable denoting total flow of *type-x* ($x \in \mathcal{X}$) into $w \in \mathcal{X} \setminus \{x\}$
f_z	:	variable denoting the maximum contribution of node z in forwarding any flow of *type-x*, $x \in \mathcal{X}$
$\delta(\mathbf{f})$	:	value of the objective function in the First Step (LP) denoting the minimum number of RNs required to be placed
p_z	:	variable denoting energy harvesting ratio ($e_z/e_{\max}$) of an RN placed at position $z \in \mathcal{Z}$
g_z	:	binary variable denoting whether z contributes to the objective value
h_z	:	variable denoting the energy harvesting ratio of an RN placed at $z \in \mathcal{Z}$ if the RN at z contributes to the objective value, and 0 otherwise.

Our unified-MILP formulation is based on multi-commodity flow-packing [27] defined on the HCG and contains two steps – *the first step* is a linear program, which seeks the placement of a minimum number of RNs, and *the second step* is a mixed integer linear program, which uses the result from the first step as one of the inputs and seeks to place the RNs in positions, such that their total energy harvesting potential is maximized. We list the notations used in our unified-MILP formulation in Table 1 and present the formulation as MILP(**f**) in Table 2 consisting of steps S1(**f**) and S2(**f**). Both formulations share some commonalities in terms of variables and notations, we present them first. In the network, we arbitrarily pick one of the base stations (say $b_1 \in \mathcal{B}$) as the common sink to which a flow of *type-x* and *value*-**f** is routed from each SN $x \in \mathcal{X}$; $\mathcal{B}' = \mathcal{B} \setminus \{b_1\}$. This *type-x* flow uses the edges of the HCG (flow f_{uvx} for $(u, v) \in$ HCG). For each $x \in \mathcal{X}$ and each $z \in \mathcal{Z}$, the amount of *type-x* flow going through node z (denoted by the variable r_{zx}) cannot exceed 1. For each $x \in \mathcal{X}$ and each $b \in \mathcal{B}'$, the amount of *type-x* flow going through node b (denoted by the variable r_{bx}) cannot exceed 1. For each $x \in \mathcal{X}$ and each $w \in \mathcal{X} \setminus \{x\}$, the amount of *type-x* flow going through node w (denoted by the variable r_{wx}) cannot exceed 1. These conditions ensure that a node other than the source can only be part of one node-disjoint path to the sink. For each $z \in \mathcal{Z}$, the *maximum amount of flow of any type* going through z is denoted by variable f_z.

First Step – S1(f): First, we will explain the constraints and then the objective function for this LP. Constraint (6.2) ensures that *the net flow of type-x out of node x is* **f**. Constraint (6.3) ensures that *the net flow of type-x into node t is* **f**. Constraints (6.4) and (6.5) ensure that for each $z \in \mathcal{Z}$ and $x \in \mathcal{X}$, the total flow of *type-x* into z is r_{zx} and that the total flow of type-x out of z is also r_{zx} – flow conservation at z. Constraints (6.6) and (6.7) ensure similar flow conservation for each $b \in \mathcal{B}'$ and $x \in \mathcal{X}$, while Constraints (6.8) and (6.9) ensure the conservation for each $w \in \mathcal{X} - \{x\}$ and $x \in \mathcal{X}$. Constraint (6.10) ensures that the flow of *type-x* on each link is a real number between 0 and 1. Constraint (6.11), together with constraints (6.4) and (6.5) ensure that for each $z \in \mathcal{Z}$ and $x \in \mathcal{X}$, the total flow of *type-x* into z is conserved and is a real number between 0 and 1.

TABLE 2
Our Unified-MILP Formulation MILP($\mathbf{f}$)

First Step – S1($\mathbf{f}$):

$$\delta(\mathbf{f}) = \mathit{min} \sum_{z\in\mathcal{Z}} f_z, \text{ over variables } f_{uvx}, r_{zx}, r_{bx}, r_{wx}, f_z \quad (6.1)$$

$$\text{s.t. } \sum_{(x,v)\in HCG} f_{xvx} - \sum_{(u,x)\in HCG} f_{uxx} = \mathbf{f}, \forall x \in \mathcal{X} \quad (6.2)$$

$$\sum_{(u,t)\in HCG} f_{utx} - \sum_{(t,v)\in HCG} f_{tvx} = \mathbf{f}, \forall x \in \mathcal{X} \quad (6.3)$$

$$\sum_{(u,z)\in HCG} f_{uzx} = r_{zx}, \forall z \in \mathcal{Z}, \forall x \in \mathcal{X} \quad (6.4)$$

$$\sum_{(z,v)\in HCG} f_{zvx} = r_{zx}, \forall z \in \mathcal{Z}, \forall x \in \mathcal{X} \quad (6.5)$$

$$\sum_{(u,b)\in HCG} f_{ubx} = r_{bx}, \forall b \in \mathcal{B}', \forall x \in \mathcal{X} \quad (6.6)$$

$$\sum_{(b,v)\in HCG} f_{bvx} = r_{bx}, \forall b \in \mathcal{B}', \forall x \in \mathcal{X} \quad (6.7)$$

$$\sum_{(u,w)\in HCG} f_{uwx} = r_{wx}, \forall w \in \mathcal{X} \setminus \{x\}, \forall x \in \mathcal{X} \quad (6.8)$$

$$\sum_{(w,v)\in HCG} f_{wvx} = r_{wx}, \forall w \in \mathcal{X} \setminus \{x\}, \forall x \in \mathcal{X} \quad (6.9)$$

$$f_{uvx} \in [0,1], \forall x \in \mathcal{X}, \forall (u,v) \in HCG \quad (6.10)$$

$$r_{zx} \in [0,1], \forall z \in \mathcal{Z}, \forall x \in \mathcal{X} \quad (6.11)$$

$$r_{bx} \in [0,1], \forall b \in \mathcal{B}', \forall x \in \mathcal{X} \quad (6.12)$$

$$r_{wx} \in [0,1], \forall w \in \mathcal{X} \setminus \{x\}, \forall x \in \mathcal{X} \quad (6.13)$$

$$r_{zx} \leq f_z, \forall z \in \mathcal{Z}, \forall x \in \mathcal{X} \quad (6.14)$$

Second Step – S2($\mathbf{f}$):

$$\max \sum_{z\in\mathcal{Z}} h_z,$$

$$\text{over variables } f_{uvx}, r_{zx}, r_{bx}, r_{wx}, f_z, g_z, h_z, \delta(\mathbf{f}) \quad (6.15)$$

s.t. All constraints (6.2) to (6.14)

$$\sum_{z\in\mathcal{Z}} f_z = \delta(\mathbf{f}) \quad (6.16)$$

$$g_z = \lceil f_z \rceil, \forall z \in \mathcal{Z} \quad (6.17)$$

$$h_z = p_z \cdot g_z, \forall z \in \mathcal{Z} \quad (6.18)$$

Constraint (6.12), (6.6), and (6.7) and Constraint (6.13), (6.8), and (6.9) ensure the same for $x \in \mathcal{X}$ and each $b \in \mathcal{B}'$ and each $w \in \mathcal{X} \setminus \{x\}$ respectively. Constraint (6.14) defines the flow packing, that is, for each $z \in \mathcal{Z}$, f_z is the *maximum* usage among all flows using z. The objective function (6.1) to be minimized is the summation of the maximum usages over all nodes $z \in \mathcal{Z}$. The objective value of S1($\mathbf{f}$) termed as $\delta(\mathbf{f})$ provides the minimum contribution of each chosen candidate location to the objective value (at most one) to ensure connectivity/survivability. However, the placement of the least number of RNs could be fractional.

Second Step – S2(f): The objective of the second step is not only to place $\delta(\mathbf{f})$ number of RNs but also to ensure that the placement is only at positions such that the sum of the energy harvesting potential of the placed RNs is maximized. Constraint (6.16) ensures that the cardinality of RNs that will be placed in a subset of positions in $\mathcal{Z}$ is equal to $\delta(\mathbf{f})$. Constraint (6.17) defines variable g_z for each $z \in \mathcal{Z}$ where g_z is zero if no RN is placed at z (no contribution to the objective value), and one otherwise. We define the energy harvesting ratio (denoted as EH-ratio) of an RN placed at position z as $p_z = e_z/e_{\max}$, where e_z is the harvesting potential at position z and $e_{\max}$ is the maximum energy harvesting potential. Consequently, in this step, the objective function maximizes the total EH-ratio of RNs placed in a subset of candidate locations under the constraint that the cardinality of the placed RNs equals $\delta(\mathbf{f})$. In essence, in this step, we are trying to place RNs in candidate locations, such that their number is minimized, and also their total energy harvesting potential is maximized. Constraints (6.16), (6.17), (6.18), and (6.2)–(6.14) ensure these conditions. An illustrative example for the MILP formulation is presented in Appendix C.

From the formulation of our unified-MILP, it is understandable that the solution would be a lower bound to the optimal in terms of the number of RNs placed (proved in Theorem 6.1). However, given that the formulation can provide fractional placement of RNs in candidate locations, the solution may not be a lower bound on the total energy harvesting potential. Given this difficulty, in the second step, our formulation finds a subset of candidate locations with total contribution $\delta(\mathbf{f})$, such that the locations have the highest total harvesting potential. The lower bound to the optimal, $\delta(\mathbf{f})$, is adequate to compare the effectiveness of our corresponding algorithms in terms of the number of RNs placed. On the other hand, the energy values from S2($\mathbf{f}$) serve as a good yardstick for comparing the effectiveness of our algorithms on the energy front, as we will show in Section 7.

Next, we prove that the solutions to MILP($\mathbf{f}$) lead to a lower bound on the corresponding optimal solutions for the CME and SME problems with respect to the number of RNs.

Theorem 6.1. Let H_1 be the number of RNs placed by a solution to MILP(1) with $\mathcal{Z}_1 \subseteq \mathcal{Z}$ being the corresponding subset of candidate locations with $f_z > 0, z \in \mathcal{Z}_1$, and H_2 be the number of RNs placed by a solution to MILP(2) with $\mathcal{Z}_2 \subseteq \mathcal{Z}$ being the corresponding subset of candidate locations with $f_z > 0, z \in \mathcal{Z}_2$, then H_1 is a lower bound to the optimal value, $Opt(1)$ of CME and H_2 is the lower bound to the optimal value, $Opt(2)$ of SME.

Proof: Consider an instance for the relay node placement problem with $R \geq r > 0$, SNs $\mathcal{X}$, BSs $\mathcal{B}$, and candidate locations $\mathcal{Z}$, with $t = b_1 \in \mathcal{B}$ being the common sink node. We will prove the connectivity case, the survivability case follows. First, let's see how $\mathcal{Z}_1$ can be used to obtain a feasible solution to the CME problem. If we place a RN at each position $z \in \mathcal{Z}_1$ for which $f(z) > 0$, then we have $Opt(1) \leq |\mathcal{Z}_1|$, hence we obtain an M-CME, that is, a feasible solution to the CME problem. On the other hand, the optimal solution to CME, $Opt(1)$, is a minimum cardinality of the RNs placed in a subset of the candidate locations such that the sum of their energy harvesting potential is high and the SNs and the BSs are connected. This placement of the RNs is in fact a feasible solution to the MILP(1) problem as $\delta(\mathbf{1}) \leq Opt(1)$. This proves that H_1 is a lower bound to the optimal solution to the CME problem $Opt(1)$. The same argument can be advanced for the relationship between $Opt(2)$ and H_2. □

7 Numerical Results

To demonstrate the effectiveness of our algorithms, in this section, we compare the results we obtain using them with those for the RNPc and RNPs problems respectively [24], and those obtained using the unified-MILP formulations (in Section 6) on the basis of the number of RNs placed and the values of the EH-ratios of the placed RNs. We denote our algorithms for the CME and SME problems as HAc (Harvesting Aware connectivity) and HAs (Harvesting Aware survivability) respectively. We denote the corresponding algorithms (RNPc and RNPs problems) in [24] as NHAc (Non-Harvesting Aware connectivity) and NHAs (Non-Harvesting Aware survivability) respectively. We denote MILPc (Mixed Integer Linear Program connectivity) as a solution to MILP(1) and MILPs (Mixed Integer Linear Program survivability) as a solution to MILP(2), and obtain these solutions using ILOG CPLEX [35]. We study the running time of the approximation algorithms, the number of RNs placed by the algorithms and the solutions of the MILPs, and also the average EH (energy harvesting) ratio of the RNs for the algorithms and the MILP solutions.

As we defined earlier, the *energy harvesting ratio (denoted as EH-ratio)* of an RN y is given by e_y/e_{max}, and it identifies harvesting efficiency of the RN, that is, the fraction of energy y can harvest. We use the *average EH-ratio*, which is the average of the EH-ratios of all deployed RNs, instead of the sum of the energy harvesting potential of each placed RN (or the sum of the EH-ratios), because the average is a better indicator of the performance of the algorithms and indicates the collective energy harvesting capability of the RNs better. The average also allows us to make fair performance comparisons – now we can compare two techniques that place disparate number of RNs as is the case of the MILP and our HA algorithms. This fact is also true if instead of the linear formulation we use the corresponding integer linear formulation. We show that the average EH-ratios of our algorithms are more than **30**% higher (in all cases) than that of the algorithms in [24]; being more than **50**% higher on an average. They perform very close to the results from the MILP formulations. Despite our approximation algorithms having higher approximation ratios, the number of RNs required are comparable to the algorithms in [24], and within 1.5 times that obtained from the MILP solutions for CME and within *twice* that obtained from the MILP solutions for SME.

For HAc and NHAc, the algorithm $\mathcal{A}$ was the 2-approximation algorithm in [18] and for HAs and NHAs, the algorithm $\mathcal{A}$ was the 3-approximation algorithm in [28], [29] – same as those used in [24]. We used C++ as the programming language and ran the numerical analysis on a 2 GHz Linux machine with 8 GB RAM.

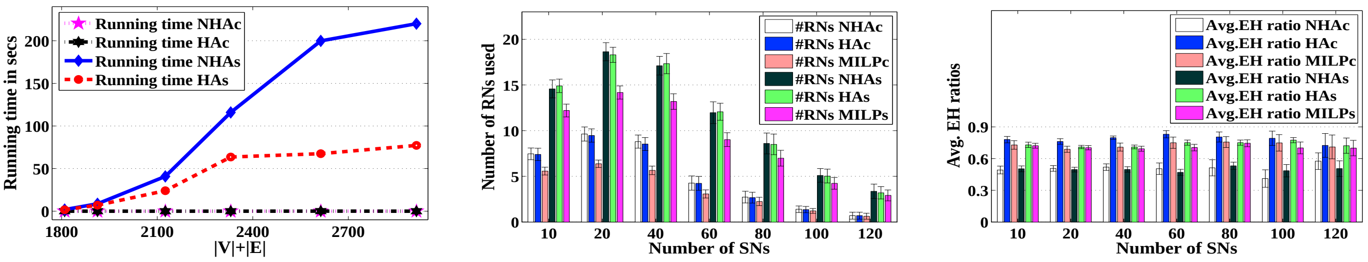
The SNs and two BSs were randomly placed in the deployment region. For the candidate locations, we used the random grid, that is, the candidate locations were randomly dispersed in the network. We chose $R = 30$m, $r = 15$m, and $e_{max} = 50$ energy-units per second. The harvesting potential $e(y)$ of an RN y was chosen randomly such that $e(y) \in \mathcal{U}[0, e_{max}]$. Based on the density of SNs in the network (#SNs/Total area), we studied the increasing density and the constant density cases. In the increasing density case, the number of possible RNs positions was fixed at 121, the number of SNs were either $10, 20, 40, 60, 80, 100$, or 120, and the field size was 100×100 sq. units. In the constant density case, we studied two densities, $d_1 = 0.005$ and $d_2 = 0.01$, with the field sizes ranging from 40×40 sq. units to 100×100 sq. units. For d_1, the number of SNs ranged from 8 to 50 and for d_2 they ranged from 16 to 100. All results were averaged over 30 runs. The graphs used in the runs, and consequently the corresponding $HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E)$, are connected (or survivable) and hence have feasible solution(s). Further, our algorithms are designed to terminate if the HCG is infeasible (refer to Steps 3 and 4 in Algorithms 1 and 2).

7.1 Increasing Density Analyses

Fig. 3 presents the results for the increasing density case in both connectivity and survivability. Fig. 3(a) presents the running times of the algorithms. As the running time depends on the number of vertices and edges of the HCG, the X-axis represents the average of the sum of the edges and vertices over the 30 instances. Both NHAc (star-thin lilac dashes) and HAc (small star-thick black dashes) take less than four seconds (two lines at the bottom). The intriguing result is that HAs (circle-red dashes) grows much slower than NHAs (diamond-blue solid). We note that the algorithm implementations for NHAs and HAs are identical, hence have no bearing on the time difference. From our analysis, we identified the difference to be a result of the weight assignment for the edges. The weight assignment for the NHA algorithms results in an edge in the HCG having a weight of either 0.5 or 1.0, whereas the assignment for HA algorithms results in edge weights in the range $[0.5, 2.0]$. Each iteration of $\mathcal{A}$ chooses an edge with the least cost that connects two sets containing a (SN-SN, SN-BS, or a BS-BS) pair that are not yet biconnected. For NHAs there is not much choice, consequently, NHAs ends up choosing a lot more unnecessary edges, which need to be deleted at the end as per $\mathcal{A}$. On the other hand, finer-grained weights in HAs allow $\mathcal{A}$ add edges more judiciously, resulting in less deletions in the end and more running time savings.

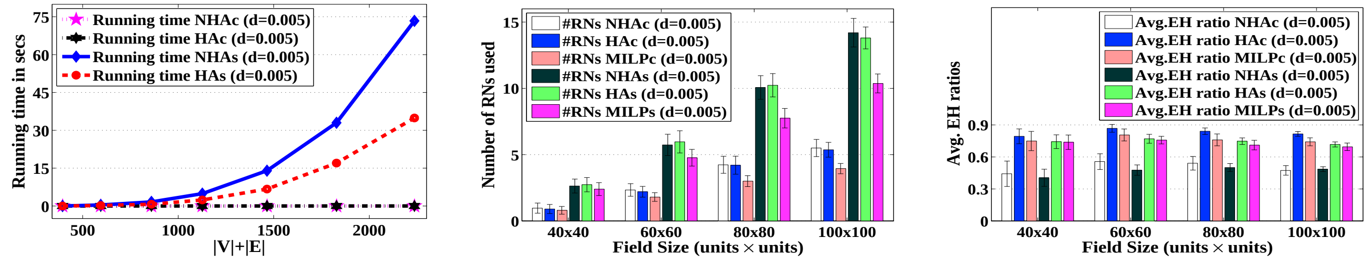
Fig. 3(b) presents the number of RNs placed by each algorithm. From the overlapping error-bars, it is clear that the number of RNs placed by HAc and HAs are comparable to that by NHAc and NHAs respectively (95% confidence interval). Despite having higher approximation ratios, in the average case, our HA algorithms perform as well as the non-HA algorithms, whose only aim is to place a minimum number of RNs. Our algorithms sometimes perform better than the NHA algorithms (20, 80, and 100) node cases, this is expected, as all algorithms are approximation algorithms and any value within the theoretical approximation bound of the optimal solution is possible. In all cases, the number of RNs required by HAc and HAs is **within** 1.5 **times** that of the number obtained by solving the corresponding MILPs, hence within 1.5 **times** the optimal. This indicates that our approximation algorithms perform very well.

Fig. 3(c) compares the average of the EH-ratios of the placed RNs – closer the ratios to 1, higher the energy harvesting potential. From the error-bars, it is clear that the results from both HA and MILP are significantly different



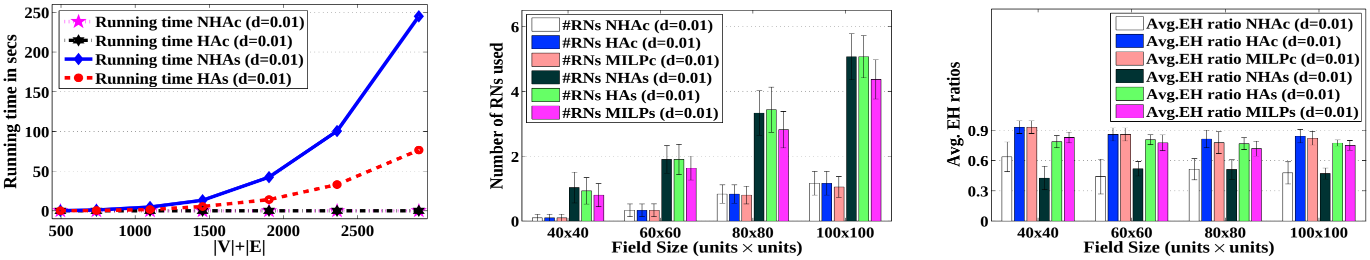
(a) Running Time (b) Number of RNs Placed (c) Average of EH Ratios

Fig. 3. Increasing Density: #RNs = 121, #SNs = $\{10, 20, 40, 60, 80, 100, 120\}$. $r = 15$m, $R = 30$m, $e_{max} = 50$ energy-units/sec.



(a) Running Time (b) Number of RNs Placed (c) Average of EH Ratios

Fig. 4. Constant Density ($d_1 = 0.005$): Field size = $\{40 \times 40, \ldots, 100 \times 100\}$, $r = 15$m, $R = 30$m, $e_{max} = 50$ energy-units/sec.



(a) Running Time (b) Number of RNs Placed (c) Average of EH Ratios

Fig. 5. Constant Density ($d_2 = 0.01$): Field size = $\{40 \times 40, \ldots, 100 \times 100\}$, $r = 15$m, $R = 30$m, $e_{max} = 50$ energy-units/sec.

(95% confidence interval) from that of NHA (especially in the survivability cases), while being similar to each other. In only one case the NHA and HA error-bars overlap: the connectivity case (NHAc/HAc), increasing density scenario with 120 SNs. This is because the density of SNs in this case is high, resulting in the network to be connected in almost 90% of sample graphs at the outset, thus needing no RNs to be placed. Thus, only a few number of sample graphs remain that have non-zero results. However, this results in inaccurate comparison. We ran HAc and NHAc for another 100 sample graphs for this case to make a more precise comparison. Among the 31 non-zero results obtained, the average EH ratios were 0.45 ± 0.09 and 0.76 ± 0.07 for NHAc and HAc respectively. This shows that even in a dense network, HA performs better (95% confidence interval).

The RNs placed by our algorithms have much better average EH-ratio than those by NHA algorithms (**ranging between 30% to 70% improvement**), which translates into more harvested energy. This implies that the placement satisfies not only the connectivity or survivability requirements, but also allows the RNs to harvest up to 70% more energy.

The averages of EH-ratios of HAc and HAs are comparable and very close to MILPc and MILPs respectively. The MILP values are a little lower in almost all cases. This is because the unified-MILP uses less number of RNs for connectivity/survivability, however, it can choose fractions of RNs (being an LP) as stipulated in S1(**f**). This does give the unified-MILP more leverage in minimizing the number of RNs that need to be placed ($\delta(\mathbf{f})$), but the use of more positions (even though fractionally) results in a bigger spread of the EH-ratios, especially lowering the minimum EH-ratio among all chosen positions, and thus a reduction in the average. On the other hand, in our algorithms the higher EH-ratios are used in identifying the positions to choose and hence the result is better.

Another reason is that the first objective for the unified-MILP is to minimize the number of RNs that need to be placed to ensure connectivity or survivability, which happens in Step S1(**f**). In Step S2(**f**), the formulation tries to place the RNs in positions that have the highest cumulative harvesting potential, while ensuring the number of RNs placed (may be fractional) is equal to $\delta(\mathbf{f})$ obtained in S1(**f**). In a sense, although this is unavoidable, Step S1(**f**) actually impedes the selection of higher EH-ratio positions by placing the constraint $\delta(\mathbf{f})$ on the number of RNs. Hence the LP suffers simultaneously from two seemingly opposite effects, thus resulting in lower average EH-ratio. We discuss more

TABLE 3
Maximum EH-ratio in Increasing Density Case

	NHAc	HAc	MILPc	NHAs	HAs	MILPs
10	0.687801	0.929979	0.99888	0.63936	0.847967	0.99888
20	0.684148	0.885283	0.99976	0.6113	0.822308	0.99991
40	0.692352	0.902571	0.99936	0.611803	0.803705	0.99936
60	0.84124	0.971575	0.99933	0.640356	0.894829	0.99933
80	0.79787	0.96571	0.99747	0.689626	0.97306	0.99747
100	0.871265	0.996175	0.99865	0.87675	0.95463	0.99834
120	0.89518	0.956925	0.99563	0.88055	0.94593	0.99747

about this in Subsection 7.3. The second reason will cause a similar disparity while comparing the ILP formulations (instead of the corresponding MILP formulations) with our algorithms.

7.2 Constant Density Analyses

Fig. 4 and Fig. 5 present the results for the constant density cases $d_1 = 0.005$ and $d_2 = 0.01$ respectively. Fig. 4(a) and Fig. 5(a) show the running time. In the increasing density case, the running time of HAs and NHAs appears to level off, but in this case the running time keeps increasing. This is because as the field size increases more sparsely spaced SNs need to be biconnected. Higher density implies more number of SNs need to be biconnected, hence the running time for the higher density case is more than the lower density case. With the increase in density, at some point, the density of sensor nodes would be high enough that we would not need RNs to connect the network or make it survivable. However, that density is higher than $d_2 = 0.01$. Note that the running time of the HA algorithms is less than that of the NHA algorithms because of the same reason as in the increasing density case.

Fig. 4(b) and Fig. 5(b) demonstrate that even in the constant density case, for both densities, our HA algorithms perform comparably to the NHA algorithms and are not penalized in terms of number of RNs. This similarity is also illustrated by the overlapping error bars. Also, the number of RNs required by our algorithms is within **1.4** times the number obtained from solving the corresponding MILP(**f**)s. This shows that our theoretical analysis on the approximation ratios of our algorithms is quite conservative – our approximation algorithms perform much better than the theoretical (worst-case) approximation ratios indicate.

Figs. 4(c) and 5(c) again demonstrate that the average EH-ratios of the RNs placed by our algorithms are much better than that of the RNs placed by the NHA algorithms. Here as well the average EH-ratio of RNs placed in our algorithm is 50% to 95% more than that of the RNs placed by the NHA algorithms. This highlights the effectiveness and efficiency of our algorithms. The EH-ratio of RNs placed using the solution to the MILP(**f**) is again lower than those of our algorithm in almost all cases. The causes are similar to the increasing density scenario – the larger search space available to the LP as well as the constraint placed by $\delta(\mathbf{f})$ in S1(**f**) on the result of S2(**f**). Due to such factors, it is required to study the maximum and minimum EH-ratios in addition to average EH-ratios.

7.3 Analyses of the Maximum and Minimum EH-ratios

For an in-depth investigation of why the MILP (as will the corresponding ILP) results in a lower average EH-ratio, in general, we need to study the difference in the maximum and minimum EH-ratios obtained by our algorithms and the solutions to the corresponding MILP formulations. For each of the 30 runs, we obtained the maximum and minimum EH-ratios of all individual RNs chosen by our algorithms and the solutions to the corresponding unified-MILPs. Tables 3, 4, and 5 show the maximum EH-ratio among the RNs chosen by NHAc, HAc, MILPc, NHAs, HAs, and MILPs, respectively, in all 30 runs, in the increasing density and both the constant density ($d_1 = 0.005$ and $d_2 = 0.01$) cases respectively.

TABLE 4
Maximum EH-ratio in Constant Density ($d_1 = 0.005$) Case

	NHAc	HAc	MILPc	NHAs	HAs	MILPs
	(d=0.005)	(d=0.005)	(d=0.005)	(d=0.005)	(d=0.005)	(d=0.005)
25x25	0.82563	0.96966	0.96966	0.857945	0.97939	0.99876
49x49	0.97252	0.99953	0.99953	0.779627	0.94625	0.99953
81x81	0.873247	0.97996	0.99946	0.660862	0.905838	0.99946
121x121	0.76917	0.94247	0.99631	0.601584	0.840018	0.99992

TABLE 5
Maximum EH-ratio in Constant Density ($d_2 = 0.01$) Case

	NHAc	HAc	MILPc	NHAs	HAs	MILPs
	(d=0.01)	(d=0.01)	(d=0.01)	(d=0.01)	(d=0.01)	(d=0.01)
25x25	0.76805	0.97942	0.97942	0.84489	0.96489	0.97942
49x49	0.91372	0.9825	0.9825	0.94543	0.99627	0.99827
81x81	0.92	0.99121	0.99121	0.89596	0.96481	0.99814
121x121	0.89204	0.9973	0.99871	0.844127	0.94205	0.99871

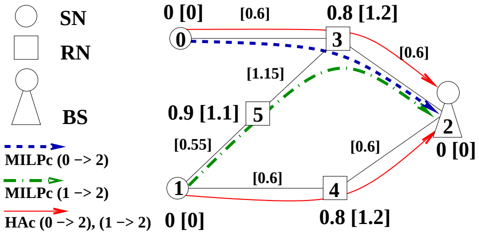


Fig. 6. Illustration for maximum EH-ratio less for solution from HAc in comparison to solution to MILPc.

From the tables, it can be observed that the *maximum* EH-ratios of the RNs placed by our algorithms (HAc/HAs) are less than or equal to the solutions to the MILP formulations (MILPc/MILPs), while being close to the corresponding values. Fig. 6 shows an illustrative example of how such scenarios occur. The graph corresponding to this case contains two SNs, $\{0, 1\}$, one BS, $\{2\}$, and three candidate locations, $\{3, 4, 5\}$. The EH-ratio of the candidate locations are given besides the locations themselves; the corresponding weights of each candidate location, calculated using Equation 3.1, are specified within the square brackets. The EH-ratios and the corresponding node weights of SNs and BS are zero. The weight of each edge in the graph, calculated using Equation 3.2, is also shown. Here, our algorithm, which selects edges based on the minimum edge weights, selects candidate locations 3 and 4 to place RNs because the positions have the least corresponding edge lengths, hence the maximum EH-ratio is 0.8 of RN 3 or 4. However, the optimal solution corresponding to the MILPc, will choose candidate locations 3 and 5, because in the first step S1(**1**) the solution is $\delta(\mathbf{1}) = 2$, the second step S2(**1**) identifies the two candidate locations (3 and 5) with the most EH-ratio (or EH potential). Therefore the maximum EH-ratio is 0.9

(RN 5) – higher than that obtained using the HAc.

From the numerical data, we also observed cases where the maximum EH-ratio from our algorithm(s) is greater than that from the corresponding MILP solution(s). This happens when the number of candidate locations ($\delta(\mathbf{f})$) obtained in Step S1($\mathbf{f}$) (and hence the unified-MILP) is less than (it can be equal to) that obtained by our HA solution and then the second step S2($\mathbf{f}$) tries to find a configuration of fractional allocation of RNs to candidate locations, so as to maximize the sum of the EH-ratios, but is restricted by $\delta(\mathbf{f})$. We illustrate our reasoning for the CME problem (as it is easier to follow) with the help of Fig. 7. An illustration of the survivability case, although non-trivial, can be created from augmenting this example.

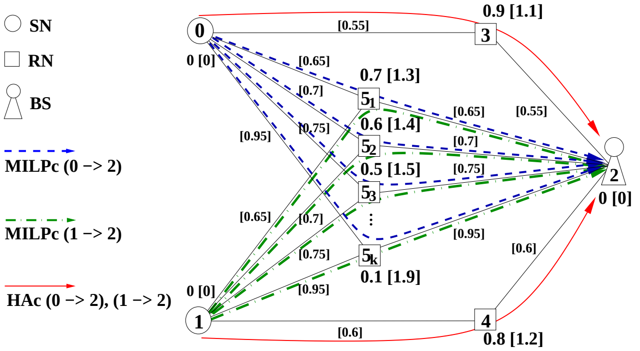


Fig. 7. Illustration for maximum EH-ratio less for solution to MILPc in comparison to solution from HAc.

In Fig. 7, the graph represents a network consisting of two SNs, $\{0, 1\}$, a BS, $\{2\}$, and a set of candidate locations, $\{3, 4, 5_1, \ldots, 5_k\}$, $k \geq 2$. The EH-ratios, weights of the nodes, and weights of the edges are calculated in the same way as for Fig. 6. We assume that $p_{5_i} \leq \min\{p_3, p_4\}, \forall 1 \leq i \leq k$. For the illustrative example in Fig. 7, our approximation algorithm HAc, will choose the two candidate locations 3 and 4 for the placement of the RNs, based on the high EH-ratios of those locations. The corresponding paths for SNs 0 and 1 to reach the BS are shown with the solid (red) arrows. However, the solution to the MILP($\mathbf{1}$) formulation, MILPc, chooses the set of locations $\{5_1, \ldots, 5_k\}$, with each location having a contribution of $1/k$ for RN placement, resulting in the placement of only one RN (split into k-portions). The dashed (blue) arrows represent the flow from SN 0 and dashed-dotted (green) arrows represent the flow from SN 1 to the BS respectively. The MILP requires only one RN to be placed, thus a better solution than our algorithm, which requires the placement of two RNs. However, the energy harvesting potentials of all these positions are less than that of 3 and 4, and so are their corresponding EH-ratios and their maximum EH-ratio.

Tables 6, 7, and 8 present the *minimum* EH-ratios among the RNs chosen by NHAc, HAc, MILPc, NHAs, HAs, and MILPs, respectively. It is very interesting to note that when the MILP solution is lower than the HA solution, it is significantly lower (several orders of magnitude in some cases). Fig. 7 illustrates one such case – it is easy to see that the minimum EH-ratio from the MILP could be arbitrarily small. Given the possibility of a larger set of nodes that the MILP can have as a solution, the possibility of one of them having a low EH-ratio is high, hence the reduced minimum EH-ratio. However, our algorithms have better minimum EH-ratio on account of the fact that they select candidate locations to minimize the edge weights chosen (refer to Algorithms 1 and 2), and hence indirectly try to choose candidate locations with high energy potential. This is a desirable, but unintended, consequence of our algorithms. Even though our objective is to find a placement such that the sum of the energy harvesting ratios are high, our algorithm also causes the minimum EH-ratio among the chosen RNs to be high as well – a desirable side-effect. These results show that indeed our algorithms are efficient and effective in achieving their desired objectives.

TABLE 6
Minimum EH-ratio in Increasing Density Case

	NHAc	HAc	MILPc	NHAs	HAs	MILPs
10	0.329132	0.567958	0.00002	0.364375	0.601911	0.00002
20	0.376629	0.585089	0.00102	0.36948	0.608629	0.00102
40	0.352475	0.709212	0.000003	0.348608	0.626374	0.00021
60	0	0	0	0.333215	0.575431	0.00041
80	0	0	0	0.318665	0.626037	0.00105
100	0	0	0	0.165265	0.449117	0.00007
120	0	0	0	0	0	0

TABLE 7
Minimum EH-ratio in Constant Density ($d_1 = 0.005$) Case

	NHAc	HAc	MILPc	NHAs	HAs	MILPs
	(d=0.005)	(d=0.005)	(d=0.005)	(d=0.005)	(d=0.005)	(d=0.005)
25x25	0	0	0	0	0	0
49x49	0	0	0	0.152095	0.49371	0.00124
81x81	0.03492	0.64745	0.00188	0.234323	0.55922	0.00015
121x121	0.25328	0.703902	0.00075	0.286628	0.571536	0.00035

We note that we do not study the ILP version of the problem because obtaining a solution for them may take much longer – sometimes months. On the other hand, the solutions to our unified-MILP formulations serve as a lower bound to the optimal number of relay nodes, and also provide useful information about the total energy harvesting potential of the RNs, for effective comparisons without incurring the penalty of time. The results help us gain a perspective on the HA solutions very quickly – solving the two MILP steps sequentially to obtain a solution takes only 2 minutes (worst-case)! Thus our MILP formulation can be used for fast comparisons even in large networks, thus is a contribution in itself.

8 Conclusions

We studied the HA-CRNP problem for connectivity and survivability. We proved the hardness of the problems and proposed a 12.4-approximation algorithm for connectivity and a 20-approximation algorithm for survivability. Because integer linear programming formulations to find the optimal solution take a long time to run, we formulated a faster running lower bound to the optimal solution using unified-MILP. We used the lower bound to demonstrate that the results of our approximation algorithms were not only

TABLE 8
Minimum EH-ratio in Constant Density ($d_2 = 0.01$) Case

	NHAc	HAc	MILPc	NHAs	HAs	MILPs
	(d=0.01)	(d=0.01)	(d=0.01)	(d=0.01)	(d=0.01)	(d=0.01)
25x25	0	0	0	0	0	0
49x49	0	0	0	0	0	0
81x81	0	0	0	0	0	0
121x121	0	0	0	0.110665	0.59914	0.00063

on average within *1.5* times that of the optimal in terms of the number of relay nodes placed but also performed favorably in terms of the energy harvesting potential of the placed RNs. In the future, we intend to explore lifetime improvements as a result of harvesting aware placement and also study the *max-max* and *max-min* versions of the problems to get deeper insights into the problem.

References

[1] A. Abu-Baker, H. Huang, E. Johnson, and S. Misra. Green Diffusion: Data Dissemination in Sensor Networks Using Solar Power. In *IEEE CCNC*, 2010.

[2] J. Bredin, E. Demaine, M. Hajiaghayi, and D. Rus. Deploying sensor networks with guaranteed capacity and fault tolerance. In *ACM MOBICOM*, pages 309–319, 2005.

[3] D. Chen, D.Z. Du, X.D. Hu, G. Lin, L. Wang, and G. Xue. Approximations for steiner trees with minimum number of steiner points. *Journal of Global Optimization*, 18:17–33, 2000.

[4] P. Cheng, C.N. Chuah, and X. Liu. Energy-aware node placement in wireless sensor networks. In *IEEE GLOBECOM*, pages 3210–3214, 2004.

[5] X. Cheng, D.Z. Du, L. Wang, and B. Xu. Relay sensor placement in wireless sensor networks. *ACM/Springer WINET*, 14(3):425–443, 2007.

[6] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs. Long-duration solar-powered wireless sensor networks. In *the 4th Workshop on Embedded Networked Sensors (EmNets 2007)*, pages 33–37, June 2007.

[7] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, second edition, 2001.

[8] K. Fan, Z. Zheng, and P. Sinha. Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks. In *ACM SenSys*, pages 239–252. ACM, 2008.

[9] L. Fleischer. A 2-approximation for minimum cost {0, 1, 2}-vertex connectivity. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 115–129, 2001.

[10] G. Gupta and M. Younis. Fault tolerant clustering of wireless sensor networks. In *IEEE WCNC*, pages 1579–1584, 2003.

[11] X. Han, X. Cao, E.L. Lloyd, and C.-C. Shen. Fault-tolerant relay node placement in heterogeneous wireless sensor networks. In *IEEE INFOCOM*, pages 1667–1675, 2007.

[12] B. Hao, J. Tang, and G. Xue. Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation. In *IEEE HPSR*, pages 246–250, 2004.

[13] Y.T. Hou, Y. Shi, H.D. Sherali, and S.F. Midkiff. Prolonging sensor network lifetime with energy provisioning and relay node placement. In *IEEE SECON*, pages 295–304, 2005.

[14] A. Kansal, J. Hsu, S. Zahedi, and M. Srivastava. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 6(32), 2007.

[15] A. Kansal and M. Srivastava. An environmental energy harvesting framework for sensor networks. In *IEEE ISLPED*, pages 481–486, 2003.

[16] A. Kashyap, S. Khuller, and M. Shayman. Relay placement for higher order connectivity in wireless sensor networks. In *IEEE INFOCOM*, pages 1–12, 2006.

[17] W. C. Ke, B. H. Liu, and M. J. Tsai. Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is np-complete. *IEEE Trans. on Computers*, 56:710–715, 2007.

[18] L.T. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica*, 15:141–145, 1981.

[19] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11:329–343, 1982.

[20] G. Lin and G. Xue. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Information Processing Letters*, 69:53–57, 1999.

[21] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava. Heliomote: Enabling long-lived sensor networks through solar energy harvesting. In *ACM SenSys (Demo Session)*, pages 309–309. ACM, 2005.

[22] H. Liu, P.J. Wan, and X.H. Jia. Fault-tolerant relay node placement in wireless sensor networks. *LNCS*, 3595:230–239, 2005.

[23] E. Lloyd and G. Xue. Relay node placement in wireless sensor networks. *IEEE Transactions on Computers*, 56:134–138, 2007.

[24] S. Misra, S. Hong, G. Xue, and J. Tang. Constrained relay node placement in wireless sensor networks: Formulation and approximations. *IEEE/ACM Transactions on Networking*, 18(2):434–448, 2010.

[25] S. Misra, N. Majd, and H. Huang. Constrained Relay Node Placement in Energy Harvesting Wireless Sensor Networks. In *8th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 25–34, 2011.

[26] J. Pan, Y.T. Hou, L. Cai, Y. Shi, and S.X. Shen. Topology control for wireless sensor networks. In *Mobicom*, pages 286–299, 2003.

[27] S.A. Plotkin, D.B. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Journal of Mathematics of Operations Research*, 20(2), 1991.

[28] R. Ravi and D. Williamson. An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica*, 18:21–43, 1997.

[29] R. Ravi and D. Williamson. Erratum: An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica*, 34:98–107, 2002.

[30] G. Robins and A. Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM J. on Disc. Math.*, 19:122–134, 2005.

[31] W. Seah, Z. Eu, and H. Tan. Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-Survey and challenges. In *Wireless Comm., Vehicular Tech., Inf. Theory & Aerospace & Electronic Sys. Tech.*, pages 1–5, 2009.

[32] R. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(23):215–233, 2003.

[33] A. Srinivas, G. Zussman, and E. Modiano. Mobile backbone networks–construction and maintenance. In *Mobihoc*, pages 166–177, 2006.

[34] J. Taneja, J. Jeong, and D. Culler. Design, modeling, and capacity planning for micro-solar power sensor networks. In *IEEE IPSN*, pages 407–418, 2008.

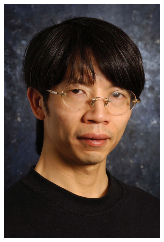
[35] 2002 [Online] ver. 8.1. Ilog cplex: Software for mathematical programming and optimization. Available: http://www.ilog.com/products/cplex/.

[36] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang. Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations. *IEEE Transactions on Mobile Computing (in press)*, 2011.

[37] W. Zhang, G. Xue, and S. Misra. Fault-tolerant relay node placement in wireless sensor networks: problems and algorithms. In *IEEE INFOCOM*, pages 1649–1657, 2007.

Satyajayant Misra (S'05) (M'09) is an Assistant Professor of Computer Science at New Mexico State University. He received his MSc degrees (2003) in physics and computer science from BITS, Pilani, India and his PhD degree (2009) in computer science from the School of Computing and Informatics at Arizona State University. His research interests are in networks, namely wireless and social networks and include network security, optimization, and analysis. He has over 30 publications in peer-reviewed international journals and conferences. He is currently serving as the TPC Vice-chair for Information Systems, IEEE INFOCOM 2012, and on the editorial boards of the IEEE Communications on Surveys and Tutorials and the IEEE Wireless Communications Magazine.

Nahid Ebrahimi Majd received her MS from Ferdowsi University, Mashhad, Iran. Currently, she is a PhD student in the department of Computer Science, New Mexico State University, Las Cruces. Her research interest is in energy harvesting wireless ad hoc networks, including relay node placement problem and cooperative caching problem in such networks.



Hong Huang (S'00) (M'03) received his B.E. degree from Tsinghua University, Beijing, China, and M.S. and Ph.D. degrees from Georgia Institute of Technology in 2000 and 2002, respectively, all in electrical engineering. He is currently an associate professor with the Klipsch School of Electrical and Computer Engineering at the New Mexico State University. His current research interests include wireless sensor networks, mobile ad hoc networks, network security, and optical networks.

Appendix A
Proofs of Lemma 4.2 and Theorem 4.2

A.1 Proof of Lemma 4.2

We note that an RN $y \in \mathcal{Y}_{opt}$ can be connected to no more than *five* SNs. This can be proved by contradiction. Let’s assume that y is connected to six SNs $x_1, \ldots x_6$. From 2-D geometry, we can see that the angle between at least two adjacent (RN-SN) edges has to be less than or equal to 60°. Without loss of generality, let the adjacent edges be (y, x_2) and (y, x_3). Since $d(y, x_2) \leq r$ and $d(y, x_3) \leq r$, hence $d(x_2, x_3) \leq r$, resulting in an edge (x_2, x_3) with weight 0. Thus we can replace edge (y, x_2) in $\mathcal{T}_{opt}$ with the edge (x_2, x_3), resulting in a tree $\mathcal{T}'_{opt}$, such that $w(\mathcal{T}'_{opt}) < w(\mathcal{T}_{opt})$, which contradicts the assumption that $\mathcal{T}_{opt}$ is an MST. Thus an RN y cannot be connected to more than five SNs in $\mathcal{T}_{opt}$.

We note that an RN $y \in \mathcal{Y}_{opt}$ can be connected to no more than *two* BSs. Let y be connected to two BSs b_1 and b_2 in $\mathcal{T}_{opt}$. As $\mathcal{T}_{opt}$ is a tree, it cannot have the zero weight edge (b_1, b_2), which would lead to a cycle. However, we can replace (y, b_2) with (b_1, b_2) to generate an MST with lesser weight than $\mathcal{T}_{opt}$, which is another contradiction.

From the above analysis, we see that an RN y can only be connected to a maximum of five SNs and one BS, a total of six. From Equation 3.1, we know that the weight of an RN is at most two and from Equation 3.2, we know that the weight of an edge is the average of the weights of the incident nodes. Hence, the weight of an edge $(y, x) \in E(\mathcal{T}_{opt})$, where $y \in \mathcal{Y}_{opt}$ and $x \in \mathcal{X} \cup \mathcal{B}$, is at most one. Therefore, for $y \in \mathcal{Y}_{opt}$ and $(y, x) \in E(\mathcal{T}_{opt})$, $\sum_y w(y, x) \leq 6$. The theorem holds for the sum of the weights of all such edges incident on $y \in \mathcal{Y}_{opt}$.

A.2 Proof of Theorem 4.2

(a) ($\Rightarrow$) If the CME problem has a feasible solution, then the set of SNs and BSs are connected, hence they have to be in the same connected component.
($\Leftarrow$) If the HCG contains $\mathcal{X} \cup \mathcal{B}$ in the same connected component, then any tree of the HCG that spans all nodes in $\mathcal{X} \cup \mathcal{B}$ is an F-CME of the instance.
(b) If the CME problem has a feasible solution, then from Lemma 4.1, we have,

$$|\mathcal{Y}_{\mathcal{A}}| \leq w(\mathcal{T}_{\mathcal{N}}) \leq w(\mathcal{T}_{\mathcal{A}}). \qquad \text{(A.1)}$$

If we denote $\mathcal{T}_{min}$ as the optimal solution to the STP instance of the HCG, it is easy to see that $w(\mathcal{T}_{\mathcal{A}}) \leq \alpha \cdot w(\mathcal{T}_{min}) \leq \alpha \cdot w(\mathcal{T}_{opt})$. Because $\mathcal{T}_{\mathcal{A}}$ is obtained by using the α-approximation algorithm for STP, whose optimal solution is $\mathcal{T}_{min}$. $\mathcal{T}_{opt}$ is a feasible solution to the STP. Hence from Lemma 4.4, we have

$$w(\mathcal{T}_{\mathcal{A}}) \leq \alpha \cdot w(\mathcal{T}_{min}) \leq \alpha \cdot w(\mathcal{T}_{opt}) \leq 8 \cdot \alpha \cdot |\mathcal{Y}_{opt}|. \qquad \text{(A.2)}$$

Combining Inequalities A.1 and A.2, $|\mathcal{Y}_{\mathcal{A}}| \leq 8 \cdot \alpha \cdot |\mathcal{Y}_{opt}|$.

Appendix B
Proofs of Lemma 5.4 and Theorem 5.2

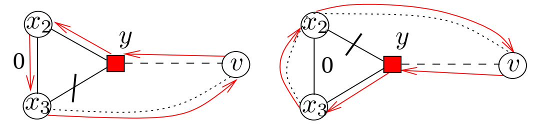
B.1 Proof of Lemma 5.4

This proof is similar to the proof of Lemma 4.1 in [24]. We note that an RN $y \in \mathcal{Y}_{opt}$ can be connected to at most *five* SNs and *one* BS respectively, in $\mathcal{H}_{opt}$. We will prove this assertion. Using contradiction, we first prove that y can only be connected to five SNs. Suppose y is an RN connected to six SNs, $x_1, \ldots, x_5$, and v. Without loss of generality, let the angle between at least two adjacent edges (y, x_2) and (y, x_3) be $\angle x_2 y x_3 \leq 60°$. This is illustrated in Fig. 8 where $x_2, x_3, v \in \mathcal{X}$ and need to be biconnected. Since $d(y, x_2) \leq r$ and $d(y, x_3) \leq r$, hence $d(x_2, x_3) \leq r$, and x_2 and x_3 can communicate directly. Since $w(x_2, x_3) = 0$, we can add the edge (x_2, x_3) to the graph.

Since $\mathcal{H}_{opt}$ is biconnected, it contains a path from x_3 to v, let’s denote this as $\pi(x_3, v)$. If $\pi(x_3, v)$ does not go through x_2, as shown in Fig. 8(a), then $\mathcal{H}_{opt}$ contains a biconnected component (cycle) composed of $\pi(x_3, v)$ and the edges (v, y), (y, x_2), and (x_2, x_3). We can remove edge (y, x_3) to obtain a biconnected subgraph $\mathcal{H}'_{opt}$ with $w(\mathcal{H}'_{opt}) < w(\mathcal{H}_{opt})$, a contradiction.

If $\pi(x_3, v)$ goes through x_2, as shown in Fig. 8(b), then $\mathcal{H}_{opt}$ contains a biconnected component (cycle) composed of $\pi(x_3, v)$ and the edges (v, y), and (y, x_3). Here, we can remove edge (y, x_2) to obtain a biconnected subgraph $\mathcal{H}'_{opt}$ with $w(\mathcal{H}'_{opt}) < w(\mathcal{H}_{opt})$, another contradiction. These two scenarios prove that an RN y can be connected to only five SNs.

We can use the same reasoning to prove that an RN y can be connected to only one BS. In Fig. 8, replace SN x_2 by the BS b_1, SN x_3 by BS b_2, and let v be an SN or an RN. The proof follows.



(a) (b)

Fig. 8. Proof of Lemma 5.4

From the above analysis, we see that an RN y can only be connected to a maximum of five SNs and one BS, a total of six. From Equation 3.1, we know that the weight of a RN is at most two and from Equation 3.2, we know that the weight of an edge is the average of the weights of the incident nodes. Hence, the weight of an edge $(y, x) \in E(\mathcal{H}_{opt})$, where $y \in \mathcal{Y}_{opt}$ and $x \in \mathcal{X} \cup \mathcal{B}$, is at most one. Therefore, for $y \in \mathcal{Y}_{opt}$ and $(y, x) \in E(\mathcal{H}_{opt})$, $\sum_y w(y, x) \leq 6$. Hence the lemma holds for the sum of the weights of all such edges incident on $y \in \mathcal{Y}_{opt}$.

B.2 Proof of Theorem 5.2

(a) ($\Rightarrow$) If the SME problem has a feasible solution, then the set of SNs and BSs are biconnected, hence they have to be in the same biconnected component.
($\Leftarrow$) If the HCG contains $\mathcal{X} \cup \mathcal{B}$ in the same biconnected component, then any biconnected subgraph of the HCG that spans all nodes in $\mathcal{X} \cup \mathcal{B}$ is an F-SME of the instance.
(b) If the SME problem has a feasible solution, then from Lemma 5.3, we have,

$$|\mathcal{Y}_{\mathcal{A}}| \leq w(\mathcal{H}_{\mathcal{N}}) \leq w(\mathcal{H}_{\mathcal{A}}). \qquad \text{(B.1)}$$

If we denote $\mathcal{H}_{min}$ as the optimal solution of the {0, 1, 2}-SNDP instance of the HCG, it is easy to see that $w(\mathcal{H}_{\mathcal{A}}) \leq$

$\alpha \cdot w(\mathcal{H}_{min}) \leq \alpha \cdot w(\mathcal{H}_{opt})$. $\mathcal{H}_{\mathcal{A}}$ is obtained by using the α-approximation algorithm for the {0, 1, 2}-SNDP problem, whose optimal solution is $\mathcal{T}_{min}$. $\mathcal{H}_{opt}$ is a feasible solution to the {0, 1, 2}-SNDP problem. Hence from Lemma 5.6, we have

$$w(\mathcal{H}_{\mathcal{A}}) \leq \alpha \cdot w(\mathcal{H}_{min}) \leq \alpha \cdot w(\mathcal{H}_{opt}) \leq 10 \cdot \alpha \cdot |\mathcal{Y}_{opt}|. \quad \text{(B.2)}$$

Combining Inequalities B.1 and B.2, $|\mathcal{Y}_{\mathcal{A}}| \leq 10 \cdot \alpha \cdot |\mathcal{Y}_{opt}|$.

Appendix C
Illustrative Example of the Unified-MILP Formulation

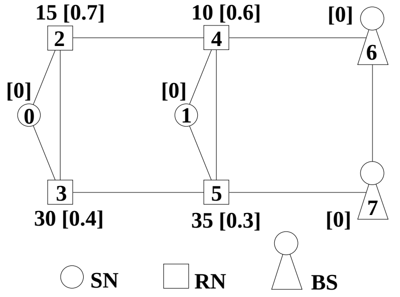


Fig. 9. Example network for the MILP formulation illustration

Fig. (9) shows a small input instance that we shall use to illustrate our formulation for both CME and SME. The instance consists of two SNs 0 and 1, four candidate locations 2, 3, 4, and 5, and two BSs 6 and 7. We pick $t = 7$ as the common sink node for our unified-MILP formulations. The numbers close to a candidate location represent the EH-potential and the EH-ratio (in square brackets) respectively; for the SNs and the BSs the corresponding values are zero. The two steps of the MILP(**f**) are illustrated in Table 9.

TABLE 9
The Unified-MILP Formulation for the Illustrative Example

S1(**f**):

$$\delta(\mathbf{f}) = min\ f_2 + f_3 + f_4 + f_5 \quad (6.1)$$

s.t.

Constraints from (6.2):

$f_{020} + f_{030} - f_{200} - f_{300} = \mathbf{f}$
$f_{141} + f_{151} - f_{411} - f_{511} = \mathbf{f}$

Constraints from (6.3):

$f_{570} + f_{670} - f_{750} - f_{760} = \mathbf{f}$
$f_{571} + f_{671} - f_{751} - f_{761} = \mathbf{f}$

Constraints from (6.4):

$f_{020} + f_{320} + f_{420} = r_{20}$
$f_{021} + f_{321} + f_{421} = r_{21}$
$f_{030} + f_{230} + f_{530} = r_{30}$
$f_{031} + f_{231} + f_{531} = r_{31}$
$f_{140} + f_{240} + f_{540} + f_{640} = r_{40}$
$f_{141} + f_{241} + f_{541} + f_{641} = r_{41}$
$f_{150} + f_{350} + f_{450} + f_{750} = r_{50}$
$f_{151} + f_{351} + f_{451} + f_{751} = r_{51}$

Constraints from (6.5):

$f_{200} + f_{230} + f_{240} = r_{20}$
$f_{201} + f_{231} + f_{241} = r_{21}$
$f_{300} + f_{320} + f_{350} = r_{30}$
$f_{301} + f_{321} + f_{351} = r_{31}$
$f_{410} + f_{420} + f_{450} + f_{460} = r_{40}$
$f_{411} + f_{421} + f_{451} + f_{461} = r_{41}$
$f_{510} + f_{530} + f_{540} + f_{570} = r_{50}$
$f_{511} + f_{531} + f_{541} + f_{571} = r_{51}$

Constraints from (6.6):

$f_{460} + f_{760} = r_{60}$
$f_{461} + f_{761} = r_{61}$

Constraints from (6.7):

$f_{640} + f_{670} = r_{60}$
$f_{641} + f_{671} = r_{61}$

Constraints from (6.8):

$f_{410} + f_{510} = r_{10}$
$f_{201} + f_{301} = r_{01}$

Constraints from (6.9):

$f_{140} + f_{150} = r_{10}$
$f_{021} + f_{031} = r_{01}$

Bound constraints from (6.10) through (6.14):

$0 \leq f_{uvx} \leq 1, \forall (u, v) \in HCG, \forall x \in \{0, 1\}$
$0 \leq r_{ux} \leq 1, \forall u \in \{2, 3, 4, 5, 6\}, \forall x \in \{0, 1\}$
$0 \leq r_{01} \leq 1$ and $0 \leq r_{10} \leq 1$
$r_{zx} \leq f_z, \forall z \in \{2, 3, 4, 5\}, \forall x \in \{0, 1\}$

S2(**f**):

$$\max\ h_2 + h_3 + h_4 + h_5 \quad (6.15)$$

s.t.

All constraints from (6.2) through (6.14)

Bound constraints from (6.16) through (6.17)

$f_2 + f_3 + f_4 + f_5 = \delta(\mathbf{f})$
$g_2 = \lceil f_2 \rceil, g_3 = \lceil f_3 \rceil, g_4 = \lceil f_4 \rceil, g_5 = \lceil f_5 \rceil$
$h_2 = 0.7g_2, h_3 = 0.4g_3,$
$h_4 = 0.6g_4, h_5 = 0.3g_5$