
Satyajayant Misra†, Member, IEEE, Nahid Ebrahim 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 consider 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 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 are 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 inter-node 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
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 BSSs; whereas, the placement for survivability ensures that the network of SNs and BSSs 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 \( O(1) \)-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 \( k = 1, 2, \) 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 \( k = 2 \), Kashyp et al. [16] presented a \( 10 \)-approximation algorithm. Bredin et al. [2] proposed polynomial time \( O(1) \)-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 \( O(1) \)-approximation algorithms for connectivity and an \( O(\log n) \)-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 \( R \geq r \), 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 \((0 \leq e_z \leq e_{\text{max}})\), where \( e_{\text{max}} \) 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,

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

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 \( \text{HCG}(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E) \), induced by the 6-tuple \((r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})\) is an edge weighted undirected graph with the vertex set \( \mathcal{V} = \mathcal{X} \cup \mathcal{B} \cup \mathcal{Z} \) and the edge set \( \mathcal{E} \) 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 \((x, y) \in \mathcal{E}\) if \( d(x, y) \leq r \). For an RN \( y \in \mathcal{Z} \) and a node \( z \in \mathcal{B} \cup \mathcal{Z} \), edge \((y, z) \in \mathcal{E}\) if \( d(y, z) \leq R \). For every pair of \( x, y \in \mathcal{B}, (x, y) \in \mathcal{E} \). 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 \( u, v \in \mathcal{Z}, 1 \leq w(u, v) \leq 2 \).

![Fig. 1. HCG with node weights](image)

(a) HCG with node weights
(b) HCG with edge weights

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 \( e_{\text{max}} = 50 \). The potentials are in energy-units per second and are the average of the 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 \( G = \text{HCG}(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}, r, R, w, E) \) 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, \( s(G) = |V(G) \cap \mathcal{Z}| \). 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,

\[
s(H) = |V(H) \cap \mathcal{Z}|. \tag{3.4}
\]
Definition 3.3. Given an undirected edge-weighted graph $G = (V, E)$, and a subset of nodes $S \subseteq V$, 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 $r(u, v)$ edge-disjoint paths between $u$ and $v$.

Also, a polynomial time $\alpha$-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 $\alpha$ 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 $\alpha$. 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. The size of the corresponding F-CME is $\max \{\sum_{e \in E} e \}$, where $e$ is $v$'s harvesting potential. The size of the corresponding F-CME is $|V|$.

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 $\sum_{e \in \mathcal{Y}} e$ is high, $e$ 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 2.4-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 $O(1)$-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 $8\alpha$-approximation algorithm for the general CME problem and a $7\alpha$-approximation algorithm for the CME problem when $\mathcal{B} = \phi$, where $\alpha$ 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 $\mathcal{Z}$, given by $\mathcal{Y}_A$.

Output: $\mathcal{F}$-CME for $(r, R, w, \mathcal{X}, \mathcal{B}, \mathcal{Z})$ given by $\mathcal{Y}_A$.

1: for $x \in \mathcal{X} \cup \mathcal{B}$, assign $c(x) = 0$ and for $z \in \mathcal{Z}$, assign $c(z) = \max_{e \in \mathcal{Y}} e + 1$.
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.
5: stop
6: apply algorithm $\mathcal{A}$ to compute a low-weight tree subgraph $T_A$ which connects all nodes in $\mathcal{B} \cup \mathcal{X}$.
7: output $\mathcal{Y}_A = Z \cap V(T_A)$

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 5–5. We apply the $\alpha$-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 \( T_A \) translates to a high energy harvesting
potential for the RNs in \( Y_A \).

![Diagram of Algorithm 1 and comparison with results of
a non-harvesting aware algorithm using an HCG with six SNs
(circles), two BSS (paw shape), and 25 candidate locations
(squares).](image)

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 (paw shape), and 25 candidate locations
(squares).

Lemma 4.1. If the CM E\((r, R, w, X, B, Z)\) has a feasible
solution, then:

(a) \( |Y_A| \leq w(T_N) \leq 2 \cdot |Y_A| \) and
(b) \( w(T_N) \leq w(T_A) \), where \( w(T_N) \) is the node-weight sum
of \( T_N \) and \( w(T_A) \) is the edge weight sum of \( T_A \).

Proof: (a) From Equation 3.1, we know that \( 1 \leq c(z) \leq 2 \),
for \( z \in Z \), and from Algorithm 1, \( Y_A \subseteq Z \). Hence we have,

\[
|Y_A| \leq \sum_{y \in Y_A} c(y) \leq 2 \cdot |Y_A| \quad (4.1)
\]

\[
w(T_N) \leq w(T_A) \leq 2 \cdot |Y_A| \quad (4.2)
\]

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 \( |Y_A| \) and \( 2 \cdot |Y_A| \).

(b) For \((u, v) \in T_{\ell N} \), \( w(u, v) = c(u) + c(v) \), hence each RN gives
half of its weight to each one of its edges. Since no RN
\( y \in T_{\ell A} \) 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 BS are 0. Therefore, \( w(T_N) \leq w(T_A) \).

Lemma 4.2. Let \( T_{\ell opt} \) be a minimum edge weighted spanning
tree (MST) of the HCG, where \( Y_{\ell opt} \) is the optimal solution to the
CM E\((r, R, w, X, B, Z)\) problem, \( y \in Y_{\ell opt} \) and
\( x \in X \cup B \) then,

\[
\sum_{(y, x) \in E(T_{\ell opt})} w(y, x) \leq 6 \cdot |Y_{\ell opt}|
\]

Proof: Refer to Appendix A.1.

Lemma 4.3. Let \( T_{\ell opt} \) be an MST of the HCG, where \( Y_{\ell opt} \)
is the optimal solution to the CM E\((r, R, w, X, B, Z)\) problem
and \( y, z \in Y_{\ell opt} \) then, \( \sum_{(y, z) \in E(T_{\ell opt})} w(y, z) \leq 2 \cdot (|Y_{\ell opt}| - 1) \).

Proof: Since \( T_{\ell opt} \) is a tree and no RN is a leaf, there are
at most \(|Y_{\ell opt}| - 1\) 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 \( 2 \cdot (|Y_{\ell opt}| - 1) \).
Lemma 4.4. Let \( T_{\text{opt}} \) be an MST of the HCG, where \( Y_{\text{opt}} \) is the optimal solution to the CME\((r, R, w, \mathcal{X}, \mathcal{B}, Z)\) problem, \( y, z \in Y_{\text{opt}} \), and \( x \in \mathcal{X} \cup \mathcal{B} \) then, \( w(T_{\text{opt}}) \leq \frac{7}{8} \cdot |Y_{\text{opt}}| \).

Proof: Since only the edges incident on the RNs contribute to the weight of \( T_{\text{opt}} \) we have,

\[
   w(T_{\text{opt}}) = \sum_{(y,z) \in E(T_{\text{opt}})} w(y,z) + \sum_{(y,x) \in E(T_{\text{opt}})} w(y,x) (4.3) 
\]

\[
   \leq 2 \cdot |Y_{\text{opt}}| - 1 + 6 \cdot |Y_{\text{opt}}| (4.4) 
\]

\[
   \leq 8 \cdot |Y_{\text{opt}}| (4.5) 
\]

Inequality 4.3 implies that the weight of \( T_{\text{opt}} \) is a sum of the weight of the edges incident on the RNs in \( Y_{\text{opt}} \). Inequality 4.4 follows from Lemmas 4.2 and 4.3.

Theorem 4.2. (a) The CME\((r, R, w, \mathcal{X}, \mathcal{B}, Z)\) 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\((r, R, w, \mathcal{X}, \mathcal{B}, Z)\) has a feasible solution, Algorithm 1 is guaranteed to find a feasible solution, which is no more than \( 8\alpha \) times the number of RNs in an optimal solution \( Y_{\text{opt}} \), where \( \alpha \) is the approximation ratio of approximation algorithm \( A \) and \( T_{\text{opt}} \) is an MST of HCG\((\mathcal{X} \cup \mathcal{B} \cup Y_{\text{opt}}, r, R, w, E)\).

Proof: Refer to Appendix A.2.

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

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

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

Proof: If we use the MST-based 2-approximation algorithm for the STP proposed in [18] as \( A \) in Algorithm 1, then the running time of Algorithm 2 for the general case will be \( O(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|^2 \log(|\mathcal{X} \cup \mathcal{B} \cup \mathcal{Z}|)) \) and the special case will be \( O(|\mathcal{X} \cup \mathcal{Z}|^2 \log(|\mathcal{X} \cup \mathcal{Z}|)) \). 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 \( HCG(\mathcal{X} \cup \mathcal{B} \cup \mathcal{Y}, r, R, w, E) \) is biconnected and \( \sum_{y \in \mathcal{Y}} \epsilon_y \) 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 \((0,1,2)\)-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 \( B = \phi \).

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 \( 10\alpha \)-approximation algorithm for the general SME problem and a \( 9\alpha \)-approximation algorithm for the SME problem when \( B = \phi \), where \( \alpha \) is the approximation ratio of an approximation algorithm \( A \) 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
Lemma 5.2. Let $|\mathcal{Y}_A|$ be an undirected biconnected 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 $\mathcal{H}_N$ as the biconnected subgraph that is derived from $\mathcal{H}_A$ obtained in Algorithm 2 by removing the edge weight $w(u,v)$ for each edge $(u,v)$ in $E(\mathcal{H}_A)$ and by reinstating the node weight $c(u)$ for each $u \in X \cup U \cup \mathcal{Y}$.

Lemma 5.3. If the $SME(r,R,w,X,B,Z)$ has a feasible solution, then:

(a) $|Y_A| \leq w(\mathcal{H}_N) \leq 2 \cdot |Y_A|$ and
(b) $w(\mathcal{H}_N) \leq w(\mathcal{H}_A)$, where $w(\mathcal{H}_N)$ is the node weights sum of $\mathcal{H}_N$ and $w(\mathcal{H}_A)$ is the edge weights sum of $\mathcal{H}_A$.

Proof: The proof follows that of Lemma 4.1.

(a) From Equation 3.1, we know that $1 \leq c(z) \leq 2$, for $z \in Z$, and from Algorithm 1, $Y_A \leq Z$. Hence we have,

$$|Y_A| \leq \sum_{y \in Y_A} c(y) \leq 2 \cdot |Y_A|$$

(b) $w(\mathcal{H}_N) \leq w(\mathcal{H}_A)$, where $w(\mathcal{H}_N)$ is the weight of all the other RNs and $x \in X \cup U$.

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

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

Proof: Refer to Appendix B.1.

Lemma 5.5. Let $\mathcal{H}_opt$ be a minimum-weight biconnected subgraph of the HCG, where $Y_{opt}$ is the optimal solution to the $SME(r,R,w,X,B,Z)$ problem and $y,z \in Y_{opt}$ then,

$$\sum_{(y,z) \in E(\mathcal{H}_{opt})} w(y,z) \leq 2 \cdot (2 \cdot |Y_{opt}| - 1)$$

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 $2|Y_{opt}| - 1$. These are the edges of the form $(y,z)$, where $y,z \in Y_{opt}$. Therefore, $\sum_{(y,z) \in E(\mathcal{H}_{opt})} w(y,z) \leq 2 \cdot (2 \cdot |Y_{opt}| - 1)$.

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

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

$$w(\mathcal{H}_{opt}) \leq \sum_{(y,z) \in E(\mathcal{H}_{opt})} w(y,z) + \sum_{(y,x) \in E(\mathcal{H}_{opt})} w(y,x) \leq 2 \cdot (2 \cdot |Y_{opt}| - 1) + 6 \cdot |Y_{opt}|$$

$$\leq 10 \cdot |Y_{opt}|$$
Inequality 5.3 implies that \( w(H_{opt}) \) is a sum of the weight of the edges incident on the RNs in \( Y_{opt} \). Inequality 5.4 follows from Lemmas 5.4 and 5.5, resulting in Inequality 5.5. 

**Theorem 5.2.** (a) The SMC(\( r, R, w, X, B, Z \)) problem has a feasible solution iff the HCG(\( X \cup B \cup Z, r, R, w, E \)) contains \( X \cup B \) in one biconnected component.

(b) If the SMC(\( r, R, w, X, B, Z \)) problem has a feasible solution, Algorithm 1 is guaranteed to find a feasible solution, which is no more than 100 times the number of RNs in an optimal solution \( Y_{opt} \), where \( \alpha \) is the approximation ratio of the approximation algorithm \( A \) and \( H_{opt} \) is a minimum-weight biconnected subgraph of HCG(\( X \cup B \cup Y_{opt}, r, R, w, E \)).

**Proof:** Refer to Appendix B.2.

**Theorem 5.3.** The general SME problem has a 20-approximation algorithm, while the special case with \( B = \emptyset \), has an 18-approximation algorithm.

**Proof:** If we use the 2-approximation algorithm proposed in [9] as algorithm \( A \), then the general case has an approximation of 10\( \alpha \), which results in a 20-approximation algorithm. When \( B = \emptyset \) the approximation ratio become 9\( \alpha \), leading to an 18-approximation algorithm. 

**Theorem 5.4.** The general SME problem has a 30-approximation algorithm with a running time of \( O(|X \cup B \cup Z|^3 + |E| \cdot |X \cup B \cup Z| \cdot \alpha(|X \cup B \cup Z|)) \), where \( E \) is the edge set of the HCG(\( X \cup B \cup Z, r, R, w, E \)). The special case with \( B = \emptyset \), has a 2\( \gamma \) approximation algorithm with a running time of \( O(|X \cup Z|^3 + |E| \cdot |X \cup Z| \cdot \alpha(|X \cup Z|)) \).

**Proof:** If we use the 3-approximation algorithm for the \( \{0, 1, 2\} \)-SNPD proposed by Ravi and Williamson [28], [29] as \( A \) 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 \( f = 1 \) corresponds to the CME problem (connectivity) and \( f = 2 \) 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.

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 \( b_1 \in B \)) as the common sink to which a flow of type-x and value-\( f \) is routed from each SN \( x \in X \); \( B' = B \setminus \{b_1\} \). This type-x flow uses the edges of the HCG (flow \( f_{ux} \) for \( (u, v) \in E \)) of HCG. For each \( x \in X \) and each \( z \in Z \), the amount of type-x flow going through node \( z \) (denoted by the variable \( r_{xz} \)) cannot exceed 1. For each \( x \in X \) and each \( b \in B' \), the amount of type-x flow going through node \( b \) (denoted by the variable \( r_{bx} \)) cannot exceed 1. For each \( x \in X \) and each \( w \in X \setminus \{x\} \), 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.

### 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 total flow of type-x into node \( t \) is \( f \). Constraints (6.4) and (6.5) ensure that for each \( z \in Z \) and \( x \in X \), the total flow of type-x into \( z \) is \( r_{xz} \) and that the total flow of type-x out of \( z \) is also \( r_{xz} \) - flow conservation at \( z \). Constraints (6.6) and (6.7) ensure similar flow conservation for each \( b \in B' \) and \( x \in X \), while Constraints (6.8) and (6.9) ensure the conservation for each \( w \in X \setminus \{x\} \) and \( x \in X \). Constraint (6.10) ensures that the flow of type-x on each link is a real number between 0 and 1.

### Table 1: Notations/Variables Used In The MILP Formulation

| \( B \) | the set of BNs  \\ | \( X \) | the set of SNs  \\ | \( Z \) | the set of candidate locations for RNs  \\ | HCG | the hybrid communication graph HCG(\( X \cup B \cup Z, r, R, w, E \))  \\ | \( f \) | \text{f = 1 for connectivity and f = 2 for survivability}  \\ | \( t \) | BS chosen as the common sink, \( t = b_1 \)  \\ | \( B' \) | the set of BSs excluding \( t = b_1 \)  \\ | \( f_{ux} \) | variable denoting type-x flow on edge \((u, v) \in E\)  \\ | \( r_{xz} \) | variable denoting total flow of type-x \((x \in X) \) into \( z \in Z\)  \\ | \( r_{bx} \) | variable denoting total flow of type-x \((x \in X) \) into \( b \in B'\)  \\ | \( r_{wx} \) | variable denoting total flow of type-x \((w \in W) \) into \( x \in X \) \}  \\ | \( g_i \) | variable denoting whether \( z \) contributes to the objective value  \\ | \( h_i \) | variable denoting the energy harvesting ratio of an RN placed at \( z \in Z \)  

If the RN at \( z \) contributes to the objective value, \( g_i = 1 \) and \( h_i = 0 \) otherwise.
TABLE 2
Our Unified-MILP Formulation MILP(\(f\))

<table>
<thead>
<tr>
<th>First Step – S1((f))</th>
<th>Second Step – S2((f))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\delta(f) = \min \sum_{x \in X} f_x) over variables (f_{ux}, r_{xz}, r_{bz}, r_{wz}, f_z, g_z, h_z, \delta(f)) (6.1)</td>
<td>(\max \sum_{x \in Z} h_x) over variables (f_{ux}, r_{xz}, r_{bz}, r_{wz}, f_z, g_z, h_z, \delta(f)) (6.15)</td>
</tr>
<tr>
<td>s.t. (\sum_{(u,t) \in HCG} f_{ux} - \sum_{(u,z) \in HCG} f_{uz} = f_x, \forall x \in X) (6.2)</td>
<td>s.t. All constraints (6.2) to (6.14)</td>
</tr>
<tr>
<td>(\sum_{(u,v) \in HCG} f_{ux} = r_{xz}, \forall z \in Z, \forall x \in X) (6.3)</td>
<td>(\sum_{z \in Z} f_x = \delta(f)) (6.16)</td>
</tr>
<tr>
<td>(\sum_{(u,v) \in HCG} f_{ux} = r_{xz}, \forall z \in Z, \forall x \in X) (6.4)</td>
<td>(g_z = [f_z]), (\forall z \in Z) (6.17)</td>
</tr>
<tr>
<td>(f_{ux} \in [0,1], \forall x \in X, (u,v) \in HCG) (6.10)</td>
<td>(h_z = p_z \cdot g_z, \forall z \in Z) (6.18)</td>
</tr>
<tr>
<td>(r_{xz} \in [0,1], \forall z \in Z, \forall x \in X) (6.11)</td>
<td></td>
</tr>
<tr>
<td>(r_{xz} \in [0,1], \forall b \in B', \forall x \in X) (6.12)</td>
<td></td>
</tr>
<tr>
<td>(r_{wz} \in [0,1], \forall w \in X \setminus {x}, \forall x \in X) (6.13)</td>
<td></td>
</tr>
<tr>
<td>(r_{xz} \leq f_z, \forall z \in Z, \forall x \in X) (6.14)</td>
<td></td>
</tr>
</tbody>
</table>

Constraint (6.12), (6.6), and (6.7) and Constraint (6.13), (6.8), and (6.9) ensure the same for \(x \in X\) and each \(b \in B'\) and each \(w \in X \setminus \{x\}\) respectively. Constraint (6.14) defines the flow packing, that is, for each \(z \in Z\), \(f_z\) 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 \(z \in Z\). The objective value of S1(\(f\)) termed as \(\delta(f)\) 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 \(\delta(f)\) 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 \(Z\) is equal to \(\delta(f)\). Constraint (6.17) defines variable \(g_z\) for each \(z \in Z\) 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 \(\delta(f)\). 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 \(\delta(f)\), such that the locations have the highest total harvesting potential. The lower bound to the optimal, \(\delta(f)\), 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(\(f\)) 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(\(f\)) 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 \(Z_1 \subseteq Z\) being the corresponding subset of candidate locations with \(f_z > 0, \forall z \in Z_1\), and \(H_2\) be the number of RNs placed by a solution to MILP(2) with \(Z_2 \subseteq Z\) being the corresponding subset of candidate locations with \(f_z > 0, \forall z \in Z_2\), then \(H_1\) is a lower bound to the optimal value, \(\text{Opt}(1)\) of CME and \(H_2\) is the lower bound to the optimal value, \(\text{Opt}(2)\) of SME.

**Proof**: Consider an instance for the relay node placement problem with \(R \geq r > 0\), SNs \(X\), BSs \(B\), and candidate locations \(Z\), with \(t = b_1 \in B\) being the common sink node. We will prove the connectivity case, the survivability case follows. First, let’s see how \(Z_1\) can be used to obtain a feasible solution to the CME problem. If we place a RN at each position \(z \in Z_1\) for which \(f(z) > 0\), then we have \(\text{Opt}(1) \leq |Z_1|\), hence we obtain an M-CME, that is, a feasible solution to the CME problem. On the other hand, the optimal solution to CME, \(\text{Opt}(1)\), 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 \(\delta(1) \leq \text{Opt}(1)\). This proves that \(H_1\) is a lower bound to the optimal solution to the CME problem \(\text{Opt}(1)\). The same argument can be advanced for the relationship between \(\text{Opt}(2)\) 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 EH-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 $A$ was the 2-approximation algorithm in [18] and for HAs and NHAs, the algorithm $A$ 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 $R = 30m$, $r = 15m$, and $e_{max} = 50$ energy-units per second. The harvesting potential $e(y)$ of an RN $y$ was chosen randomly such that $e(y) \in U(0, e_{max})$. 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 $100 \times 100$ 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 $40 \times 40$ sq. units to $100 \times 100$ 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($X \cup B \cup Z, r, R, w, E$), 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 implementation 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 HAc 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 $A$ 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 bi-connected. 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 $A$. On the other hand, finer-grained weights in HAs allow $A$ 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 HAc 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.
energy. But also allows the RNs to harvest up to more harvested energy. This implies that the placement remain that have non-zero results. However, this results in comparison. Among the ranging average EH-ratios were 0.0$\pm$0.01 non-zero results obtained, the constraint $|V|+|E|= 0$ of RNs (being an LP) as stipulated in S1($f$). In Step S2($f$), the use of obtained in S1($f$)) actually suffers simultaneously from two seemingly opposite effects, thus resulting in lower average EH-ratio. We discuss more another reason is that the first objective for the unified-MILP is to minimize the number of RNs that need to be placed (δ($f$)), 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.

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 (δ($f$)), 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 δ($f$) 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 δ($f$) 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
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(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 $\delta(f)$ 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 well as 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.

![Fig. 6. Illustration for maximum EH-ratio less for solution from HAc in comparison to solution to MILPc.](image)

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 $\delta(1) = 2$, 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 ($\delta(f)$) obtained in Step S1(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(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 $\delta(f)$. 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(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.

We note that we do not study the IP 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 6

<table>
<thead>
<tr>
<th>NHAc</th>
<th>HAc</th>
<th>MILPc</th>
<th>NHAc</th>
<th>HAc</th>
<th>MILPc</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.329132</td>
<td>0.567958</td>
<td>0.00002</td>
<td>0.364375</td>
<td>0.601911</td>
</tr>
<tr>
<td>20</td>
<td>0.376629</td>
<td>0.585089</td>
<td>0.00012</td>
<td>0.369488</td>
<td>0.606826</td>
</tr>
<tr>
<td>40</td>
<td>0.352475</td>
<td>0.709212</td>
<td>0.00003</td>
<td>0.384608</td>
<td>0.626374</td>
</tr>
<tr>
<td>60</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.332315</td>
<td>0.575431</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.318665</td>
<td>0.626037</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.165265</td>
<td>0.449117</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 7

<table>
<thead>
<tr>
<th>NHAc (d=0.005)</th>
<th>HAc (d=0.005)</th>
<th>MILPc (d=0.005)</th>
<th>NHAc (d=0.005)</th>
<th>HAc (d=0.005)</th>
<th>MILPc (d=0.005)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25x25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>49x49</td>
<td>0</td>
<td>0</td>
<td>0.152095</td>
<td>0.493710</td>
<td>0.507815</td>
</tr>
<tr>
<td>81x81</td>
<td>0.03492</td>
<td>0.64745</td>
<td>0.00188</td>
<td>0.234323</td>
<td>0.59922</td>
</tr>
<tr>
<td>121x121</td>
<td>0.25328</td>
<td>0.703902</td>
<td>0.000075</td>
<td>0.286628</td>
<td>0.571536</td>
</tr>
</tbody>
</table>

TABLE 8

<table>
<thead>
<tr>
<th>NHAc (d=0.01)</th>
<th>HAc (d=0.01)</th>
<th>MILPc (d=0.01)</th>
<th>NHAc (d=0.01)</th>
<th>HAc (d=0.01)</th>
<th>MILPc (d=0.01)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25x25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>49x49</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>81x81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>121x121</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Minimum EH-ratio in Constant Density (d = 0.01) Case
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


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 include 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 $y \in \mathcal{Y}_{\text{opt}}$ 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 $T_{\text{opt}}$ with the edge $(x_2, x_3)$, resulting in a tree $T'_{\text{opt}}$, such that $w(T'_{\text{opt}}) < w(T_{\text{opt}})$, which contradicts the assumption that $T_{\text{opt}}$ is an MST. Thus an RN $y$ cannot be connected to more than five SNs in $T_{\text{opt}}$.

We note that an RN $y \in \mathcal{Y}_{\text{opt}}$ can be connected to no more than two BSs. Let $y$ be connected to two BSs $b_1$ and $b_2$ in $T_{\text{opt}}$. As $T_{\text{opt}}$ 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 $T_{\text{opt}}$, 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 $(y, x) \in E(T_{\text{opt}})$, where $y \in \mathcal{Y}_{\text{opt}}$ and $x \in \mathcal{X} \cup \mathcal{B}$, is at most one. Therefore, for $y \in \mathcal{Y}_{\text{opt}}$ and $(y, x) \in E(T_{\text{opt}})$, $\sum_y w(y, x) \leq 6$. The theorem holds for the sum of the weights of all such edges incident on $y \in \mathcal{Y}_{\text{opt}}$.

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,

$$|\mathcal{Y}_{\text{A}}| \leq w(T_{\mathcal{X}}) \leq w(T_{\mathcal{A}}). \tag{A.1}$$

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

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

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

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 $y \in \mathcal{Y}_{\text{opt}}$ can be connected to at most five SNs and one BS respectively, in $\mathcal{H}_{\text{opt}}$. 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 $\angle x_2yx_3 \leq 60°$. 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 $\mathcal{H}_{\text{opt}}$ 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 $\mathcal{H}_{\text{opt}}$ contains a biconnected component (cycle) composed of $\pi(x_3, v)$ and the edges $(y, x_2)$, $(y, x_3)$, and $(x_2, x_3)$. We can remove edge $(y, x_3)$ to obtain a biconnected subgraph $\mathcal{H}'_{\text{opt}}$ with $w(\mathcal{H}'_{\text{opt}}) < w(\mathcal{H}_{\text{opt}})$, a contradiction.

If $\pi(x_3, v)$ goes through $x_2$, as shown in Fig. 8(b), then $\mathcal{H}_{\text{opt}}$ contains a biconnected component (cycle) composed of $\pi(x_3, v)$ and the edges $(v, y)$, $(y, x_3)$, and $(x_2, x_3)$. Here, we can remove edge $(y, x_3)$ to obtain a biconnected subgraph $\mathcal{H}'_{\text{opt}}$ with $w(\mathcal{H}'_{\text{opt}}) < w(\mathcal{H}_{\text{opt}})$, 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.

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 $(y, x) \in E(\mathcal{H}_{\text{opt}})$, where $y \in \mathcal{Y}_{\text{opt}}$ and $x \in \mathcal{X} \cup \mathcal{B}$, is at most one. Therefore, for $y \in \mathcal{Y}_{\text{opt}}$ and $(y, x) \in E(\mathcal{H}_{\text{opt}})$, $\sum_y w(y, x) \leq 6$. Hence the lemma holds for the sum of the weights of all such edges incident on $y \in \mathcal{Y}_{\text{opt}}$.

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,

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

If we denote $\mathcal{H}_{\text{min}}$ as the optimal solution of the $\{0, 1, 2\}$-SNDP instance of the HCG, it is easy to see that $w(\mathcal{H}_{\mathcal{A}}) \leq \alpha \cdot w(\mathcal{H}_{\mathcal{A}})$.
\[ \alpha \cdot w(\mathcal{H}_{\text{min}}) \leq \alpha \cdot w(\mathcal{H}_{\text{opt}}). \] \[ \mathcal{H}_A \text{ is obtained by using the } \alpha\text{-approximation algorithm for the } \{0, 1, 2\}\text{-SNDP problem, whose optimal solution is } T_{\text{min}}. \mathcal{H}_{\text{opt}} \text{ is a feasible solution to the } \{0, 1, 2\}\text{-SNDP problem. Hence from Lemma 5.6, we have} \]
\[ w(\mathcal{H}_A) \leq \alpha \cdot w(\mathcal{H}_{\text{min}}) \leq \alpha \cdot w(\mathcal{H}_{\text{opt}}) \leq 10 \cdot \alpha \cdot |Y_{\text{opt}}|. \] (B.2)

Combining Inequalities B.1 and B.2, \[ |Y_A| \leq 10 \cdot \alpha \cdot |Y_{\text{opt}}|. \]

**APPENDIX C**

**ILLUSTRATIVE EXAMPLE OF THE UNIFIED-MILP FORMULATION**

![Example network for the MILP formulation illustration](image)

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**

<table>
<thead>
<tr>
<th>S1(( f ))</th>
<th>( \delta(( f )) = \min f_2 + f_3 + f_4 + f_5 ) (6.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s.t.</td>
<td>Constraints from (6.2):</td>
</tr>
<tr>
<td></td>
<td>( f_{020} + f_{030} + f_{200} + f_{300} = f_2 )</td>
</tr>
<tr>
<td></td>
<td>( f_{141} + f_{151} + f_{141} + f_{151} = f_3 )</td>
</tr>
<tr>
<td></td>
<td>Constraints from (6.3):</td>
</tr>
<tr>
<td></td>
<td>( f_{570} + f_{670} + f_{750} + f_{760} = f_4 )</td>
</tr>
<tr>
<td></td>
<td>( f_{571} + f_{671} + f_{751} + f_{761} = f_5 )</td>
</tr>
<tr>
<td></td>
<td>Constraints from (6.4):</td>
</tr>
<tr>
<td></td>
<td>( f_{200} + f_{230} + f_{240} + f_{240} = r_{20} )</td>
</tr>
<tr>
<td></td>
<td>( f_{021} + f_{321} + f_{421} + f_{421} = r_{21} )</td>
</tr>
<tr>
<td></td>
<td>( f_{030} + f_{230} + f_{350} + f_{350} = r_{30} )</td>
</tr>
<tr>
<td></td>
<td>( f_{031} + f_{231} + f_{351} + f_{351} = r_{31} )</td>
</tr>
<tr>
<td></td>
<td>( f_{140} + f_{140} + f_{540} + f_{540} = r_{40} )</td>
</tr>
<tr>
<td></td>
<td>( f_{141} + f_{141} + f_{541} + f_{541} = r_{41} )</td>
</tr>
<tr>
<td></td>
<td>( f_{150} + f_{350} + f_{540} + f_{540} = r_{50} )</td>
</tr>
<tr>
<td></td>
<td>( f_{151} + f_{551} + f_{541} + f_{541} = r_{51} )</td>
</tr>
<tr>
<td></td>
<td>Constraints from (6.5):</td>
</tr>
<tr>
<td></td>
<td>( f_{460} + f_{760} = r_{60} )</td>
</tr>
<tr>
<td></td>
<td>( f_{461} + f_{761} = r_{61} )</td>
</tr>
<tr>
<td></td>
<td>Constraints from (6.6):</td>
</tr>
<tr>
<td></td>
<td>( f_{640} + f_{670} = r_{60} )</td>
</tr>
<tr>
<td></td>
<td>( f_{641} + f_{671} = r_{61} )</td>
</tr>
<tr>
<td></td>
<td>Constraints from (6.7):</td>
</tr>
<tr>
<td></td>
<td>( f_{410} + f_{510} = r_{10} )</td>
</tr>
<tr>
<td></td>
<td>( f_{201} + f_{301} = r_{01} )</td>
</tr>
<tr>
<td></td>
<td>Constraints from (6.8):</td>
</tr>
<tr>
<td></td>
<td>( f_{140} + f_{150} = r_{10} )</td>
</tr>
<tr>
<td></td>
<td>( f_{021} + f_{031} = r_{01} )</td>
</tr>
<tr>
<td></td>
<td>Bound constraints from (6.10) through (6.14):</td>
</tr>
<tr>
<td></td>
<td>( 0 \leq f_{ux} \leq 1, \forall (u,v) \in HCG, \forall x \in {0, 1} )</td>
</tr>
<tr>
<td></td>
<td>( 0 \leq r_{ux} \leq 1, \forall u \in {2, 3, 4, 5, 6}, \forall x \in {0, 1} )</td>
</tr>
<tr>
<td></td>
<td>( 0 \leq r_{01} \leq 1 \text{ and } 0 \leq r_{10} \leq 1 )</td>
</tr>
<tr>
<td></td>
<td>( r_{ux} \leq f_z, \forall x \in {2, 3, 4, 5}, \forall x \in {0, 1} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S2(( f ))</th>
<th>( \max h_2 + h_3 + h_4 + h_5 ) (6.15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s.t.</td>
<td>All constraints from (6.2) through (6.14)</td>
</tr>
<tr>
<td></td>
<td>Bound constraints from (6.16) through (6.17)</td>
</tr>
<tr>
<td></td>
<td>( f_2 + f_3 + f_4 + f_5 = \delta(( f )) )</td>
</tr>
<tr>
<td></td>
<td>( g_2 = [f_2], g_3 = [f_3], g_4 = [f_4], g_5 = [f_5] )</td>
</tr>
<tr>
<td></td>
<td>( h_2 = 0.7g_2, h_3 = 0.4g_3, )</td>
</tr>
<tr>
<td></td>
<td>( h_4 = 0.6g_4, h_5 = 0.3g_5 )</td>
</tr>
</tbody>
</table>