Signal-Aware Green Wireless Relay Network Design

Chenfei Gao*, Jian Tang*, Xiang Sheng*, Weiyi Zhang†, Chonggang Wang‡

* Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244
† AT&T Labs Research, Middletown, NJ 07748
‡ InterDigital Communications, King of Prussia, PA 19406

Abstract—Small cell network is the new trend for next generation mobile network design. One feasible model is using Relay stations (RS) as small cell providers to achieve extended coverage, lower cost, and higher network capacity. This paper studies Signal-aware relay station placement and power allocation problem in wireless relay networks with multiple base stations in the field. This problem consists of both subscriber coverage problem and relay power optimization problem, which have not been extensively studied together in previous works. This work takes into account physical constraints such as channel capacity, signal to noise ratio (SNR) requirement of subscribers, relay power cost and network topology. We set up a two-step goal that is firstly to find minimum number of RS in order to cover all the subscribers meeting each SNR requirement, and then to ensure communications built between any subscriber to a base station. In order to ensure each subscriber’s SNR, transmission power of each RS should be adjustable. Thus, minimizing power cost of RSs is our goal in the second step. We divide the problem into two sub-problems, Lower-tier Coverage Relay Allocation (LCRA) problem and Upper-tier Connectivity Relay Allocation (UCRA) problem. For the LCRA problem, we present two approximation solutions based on minimum hitting set and maximum independent set. For the UCRA problem, an approximation algorithm and an optimal algorithm are proposed. At the end, an approximation solution for our original problem, which combines the approaches of the two sub-problems, is provided. Numerical results are presented to confirm the theoretical analysis of our schemes, and to show strong performances of our solutions.

I. INTRODUCTION

With mobile operators struggling to support the growth in mobile data traffic, many are using mobile data offloading as a more efficient use of radio spectrum. Small cells is a new trend for next generation wireless networks since many mobile network operators see small cells as vital to managing spectrum more efficiently. Hence, small cell network scheme can help the network carriers to achieve extended coverage and higher network capacity. One of the feasible small cell network design is using Relay Stations (RS) to offload traffic that directly transmitted to/from macro cells.

Relay station placement has been an active research topic in wireless networks, especially in wireless sensor networks. By using RSs, one could deploy a network at a lower cost than using only (more expensive) BSs to provide wide coverage while delivering a required level of service to users [7]–[10]. In [11], Lin and Xue proved the single-tiered placement problem with \( R = r \) and \( K = 1 \) is NP-hard, where \( R \), \( r \) and \( K \) denote the transmission range of relay nodes, the transmission range of sensor nodes, the connectivity requirement respectively. A 5-approximation algorithm was presented to solve the problem. The authors also designed a steinerization scheme which has been used by many later works. Beside minimizing the number of placed RSs, work also has been done on placement with physical constraints, such as energy consumption and network lifetime. Hou et al. studied the energy provisioning problem for a two-tiered wireless sensor network [12]. Besides provisioning additional energy on the existing nodes, they consider deploying relay nodes (RNs) into the network to mitigate network geometric deficiency and prolong network lifetime. In [13], Hassanein et al. proposed three random relay deployment strategies for connectivity-oriented, lifetime-oriented and hybrid deployment. In [14], Pan et al. studied base station placement to maximize network lifetime. Recently, a new dual-relay coverage architecture was proposed for 802.16j Mobile Multi-hop Relay-based (MMR) networks [8], [9], where each subscriber station (SS) is covered by two RSs. [8] assumed that only one RS is placed in each cell. ILP formulation was applied to find an optimal placement of RS which can maximize the cell capacity in terms of user traffic rates. In [9], assuming a uniform distribution on user traffic demand, the authors studied how to determine the RSs’ locations from a set of predefined candidate positions. The authors of [1] studied multiple hop relay problem with consideration of channel capacity. Two tiers model was mentioned as well, but it addressed the relay placement problem on condition that all relay nodes forwarded traffic in their maximum transmission power. In addition, an efficient MUST algorithm was proposed to address the connectivity problem on upper tier. However, MUST worked under the physical constraint of only one base station in the field. In this paper, we improve the research of [1] pretty much by eliminating several assumptions and physical constraints in their work and we take subscriber’s SNR constraint into account in practical way. To the best of our knowledge, this paper is the first one to study low-cost multiple hop relay problem considering channel capacity, subscriber’s SNR requirement, power consumption of relay nodes and allowing multiple base stations existing in the topology of wireless relay networks.

The rest of the paper is organized as follows. Section II provides concept definitions and problem statement. Section III presents Linear Programming with Quadratic Constraints...
(LPQC) based solutions and approximation algorithms. In Section IV, we use extensive numerical results to show the good performances of our proposed schemes. This paper is concluded in section V.

II. PROBLEM STATEMENTS

In our model, a wireless relay network consists of Subscriber Stations (SS), Base Stations (BS), and Relay Stations. In reality, several types of SS exist, including static SS, ad-hoc SS and compound SS. In this work, we assume that SSs are static users such as Wal-mart, McDonald’s, and gas stations, which are static but have large traffic demands. Similarly, all the RSs, with the function of relaying traffic coming from BS, other RSs, or SS, are assumed to be fixed as well in this paper. Our two-tier network model divides the network into two tiers, lower tier and upper tier. On lower tier, coverage RSs are placed in order to cover all the SSs while meeting SS’s performance requirements such as channel capacity, SNR threshold. The communications on lower tier is mainly between SSs and coverage RSs. We name these communications “access links”. On upper tier, connectivity RSs are to be placed in order to connect coverage RSs to BSs, using possible multiple-hop relay model. The communication links on the upper tier are denoted as “relay links” in this paper. The scenario described above is illustrated in Fig. 1.

A. SNR-Aware Green Relay Allocation

Each SS needs to be covered by an RS or BS for traffic transmission. Different from most previous work, we take channel capacity and SNR threshold into consideration in this work. The access links between an SS and its coverage RS should provide enough channel capacity to satisfy the SS’s data rate request. In addition, for each SS being able to correctly decode signals, its received Signal to Noise ratio (SNR) is another parameter that should be considered. For instance, SNR received by a mobile phone is usually in the range of $-10$dB to $-20$dB so that we can correctly decode the signals. Since all the SSs are in our work sharing similar communication purposes and QoS, we assume that all SSs have the same SNR threshold.

**Definition 1** (Feasible coverage). Let $s_i$ be a fixed SS with known location, and $b_i$ be its data rate request (in terms of bps). An RS $r_m$ is said to provide a feasible coverage for $s_i$ if the channel capacity of the link (in terms of bps) between $s_i$ and $r_m$ is sufficient for the data rate request of $s_i$; and, the SNR received at $s_i$ is above the SNR threshold.

**Definition 2** (SNR for subscribers). Let $s_i$ be an SS, $R = \{r_1, r_2, ..., r_n\}$ be the RS set and $P = \{p_1, p_2, ..., p_n\}$ be the set of received power by $s_i$ from each RS. If SS $s_i$ receives signal from RS $r_j$, the SNR at $s_i$ is defined as $\frac{p_j}{\sum_{r_k \in R} p_k}$. □

To simplify the study, we transform the capacity and SNR requirements into distance requirements since the capacity of a wireless link is highly related to the distance between transmitters and receivers [3]. In this paper, we choose two-ray ground path loss model for modeling the long distance LOS channel with large scale signal strength. The received power $P_r$ is given as

$$P_r = P_t G_t G_r h_t^2 h_r^2 d^{-\alpha}$$  \hspace{1cm} (2.1)

where $P_t$ is the transmission power, and $G_t, G_r$ and $h_t, h_r$ are the gains and heights of transmitter tower and receiver tower, respectively, $d$ is the Euclidean distance between the two end nodes, $\alpha$ is the attenuation factor, which usually

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td>fixed subscriber station $i$ with known location</td>
</tr>
<tr>
<td>$b_i$</td>
<td>data rate request of subscriber station $i$</td>
</tr>
<tr>
<td>$r_j$</td>
<td>relay station $j$ with known location</td>
</tr>
<tr>
<td>$d_i$</td>
<td>feasible coverage distance of subscriber station $i$</td>
</tr>
<tr>
<td>$c_i$</td>
<td>feasible coverage circle of subscriber station $i$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>SNR threshold of each subscriber</td>
</tr>
<tr>
<td>$T_{ij}$</td>
<td>indicator that denotes if position $j$ is chosen to be placed RS in $s_i$</td>
</tr>
<tr>
<td>$n$</td>
<td>the total number of subscribers</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>the distance between position $j$ and position $i$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>the attenuation parameter</td>
</tr>
<tr>
<td>$S$</td>
<td>set of all the subscribers</td>
</tr>
<tr>
<td>$D$</td>
<td>set of all the feasible coverage distances of subscribers</td>
</tr>
<tr>
<td>$L_{ss}$</td>
<td>set of independent SS groups obtained from Algorithm 2</td>
</tr>
<tr>
<td>$L_{ss}'$</td>
<td>set of independent SS groups obtained from Algorithm 2</td>
</tr>
<tr>
<td>$L_{RS}$</td>
<td>set of all coverage RS groups placed for each SS group</td>
</tr>
<tr>
<td>$L_{RS}'$</td>
<td>set of all coverage RS groups placed for independent SS group $i$</td>
</tr>
<tr>
<td>$K_{mhs}$</td>
<td>set of minimum hitting sets of coverage RSs for all SS groups</td>
</tr>
<tr>
<td>$K_{mhs}^i$</td>
<td>minimum hitting set of coverage RSs for independent SS group $i$</td>
</tr>
<tr>
<td>$G_i$</td>
<td>coverage link pair for independent SS group $i$</td>
</tr>
<tr>
<td>$N_{max}$</td>
<td>maximum noise which can be ignored</td>
</tr>
<tr>
<td>$P_{max}$</td>
<td>the maximum transmission power of relay station</td>
</tr>
<tr>
<td>$d_{max}$</td>
<td>the maximum distance above which the noise can be ignored</td>
</tr>
<tr>
<td>$d_{eff}$</td>
<td>the effective noise influence distance</td>
</tr>
<tr>
<td>$P_t$</td>
<td>the transmission power of the BS in coverage RSs</td>
</tr>
<tr>
<td>$P_{rs}$</td>
<td>set of minimum received powers requested by each SS</td>
</tr>
<tr>
<td>$L_{vec}$</td>
<td>set of coverage RSs placed on lower tier</td>
</tr>
<tr>
<td>$L_{high}$</td>
<td>set of connectivity RSs placed on upper tier</td>
</tr>
<tr>
<td>$P_{c}$</td>
<td>the coverage power of $P_c$, for RS $r_i$</td>
</tr>
<tr>
<td>$P_{SNR}$</td>
<td>the SNR power $P_{SNR}$ for RS $r_i$</td>
</tr>
<tr>
<td>$R_c$</td>
<td>set of coverage RSs</td>
</tr>
<tr>
<td>$D_c$</td>
<td>set of distance requirements of $R_c$</td>
</tr>
<tr>
<td>$d_{c}$</td>
<td>distance requirement of coverage RS $r_i$</td>
</tr>
<tr>
<td>$B$</td>
<td>set of base stations</td>
</tr>
<tr>
<td>$b_{si}$</td>
<td>base station $i$</td>
</tr>
<tr>
<td>$P_{t}$</td>
<td>sum of transmission power of coverage RSs on lower tier</td>
</tr>
<tr>
<td>$P_{total}$</td>
<td>sum of transmission power of all placed RSs</td>
</tr>
</tbody>
</table>

**TABLE I**: List of all the symbols used in this paper
varies in a range of 2−4. The signal-to-noise ratio (SNR) at receiver is $SNR_r = P_r/N_0$, where $N_0$ is the thermal noise at the receiver. We could see it as a constant. According to Shannon’s theorem, wireless link capacity is given by $C = B \log(1 + SNR_r)$, where $B$ is the channel bandwidth. Thus, if noise $N_0$ is a constant, the channel capacity (in terms of bps) is only related to the received signal power $P_r$ and moreover only related to the distance between two end nodes assuming transmission power $P_t$ of RS is constant. Therefore, the capacity requests of SS are equivalent to distance requests between SS and its corresponding RS. Currently, we can say, SS $s_i$ is covered by RS $r_j$ if and only if the distance $d(s_i, r_j)$ is less or equal to SS’s distance request $d_i$.

Unlike previous coverage problems, which assume that RSs always transmit in maximum power, we can adjust power consumption of RSs as long as not changing the coverage topology. Therefore, after the placement problem assuming that RSs using maximum transmission power, we further study a power cost optimization problem. Our proposed SNR Aware Green (SAG) Relay problem is to find the minimum number of RSs to ensure SS traffic while minimizing the power consumption of the placed RSs.

**Definition 3 (SNR Aware Green (SAG) Relay problem).** Given a wireless relay network with multiple BSs and a set of SSs $S = \{s_1, s_2, ..., s_n\}$, let $D = \{d_1, d_2, ..., d_n\}$ and $SNR_i$ be the feasible distance requests and SNR threshold for SSs, respectively. The SAG problem seeks a minimum number of RSs $R$ and transmission power allocation strategy for $R$ such that:

1. Providing feasible coverage for each $s_i \in S$. In other words, each SS $s_i \in S$ has enough SNR and an access link with enough capacity to an RS or BS;
2. Each placed RS must enough capacity on relay links to relay traffic to a BS;
3. Sum of transmission power of the placed RS should be minimized.

**III. APPROXIMATION SOLUTIONS FOR SAG**

Our SAG problem consists of two aspects, coverage with minimum number of RSs and minimization of transmission power of the placed RSs. A similar problem, DARP, has been studied in [1] without considering power minimization. Since DARP is estimated to be NP-hard [1], we expected SAG to be NP-hard as well. To solve SAG, we divide it into two sub-problems, Lower-tier Coverage Relay Allocation (LCRA) problem and Upper-tier Connectivity Relay Allocation (UCRA) problem, and try to tackle them one by one. A solution for SAG will be presented based on LCRA and UCRA in next section.

**A. Lower-tier Coverage Relay Allocation (LCRA) problem**

First, we assume that all the RSs are operating with maximum transmission power. With this assumption, we aim to find a minimum number of RSs to provide feasible coverage for all the SSs.

**Definition 4 (Lower-tier Coverage Relay Allocation (LCRA) problem).** Given a wireless relay network with a set of subscriber station $S = \{s_1, s_2, ..., s_n\}$, let $D = \{d_1, d_2, ..., d_n\}$ be the distance requirement set for SSs, and $SNR_i$ be the SNR threshold. The LCRA problem seeks a minimum number of relay stations $R$ to provide feasible coverage for $s_i \in S$, and the total transmission power by RSs in $R$ is minimized.

Unlike the pure coverage problems, LCRA problem takes SS’s SNR requests into consideration, which makes the LCRA problem more complicated. For example, to solve pure coverage problems in [1], we could allocate circles’ intersection points as candidate positions, which are finite, to find best solutions. However, circles’ intersection points cannot even guarantee feasible solutions for LCRA due to the SNR requirements. If we put multiple RSs to cover multiple SSs, the distance requirement of SSs could be satisfied. However, it is possible that RSs could interfere signals from each other, and results in unbearable SNR at some SSs. To find a more appropriate solution for feasible coverage, we propose to use small scale of grids spreading around the entire field as candidate RS locations. The benefit of using grids is that nearly every place in the field can be considered if we adjust the grid size small enough. The issue is that smaller grid size will make more candidate positions. The running time is non-linearly increasing with the number of grid candidates. Therefore, how to pick a right grid size to achieve the best tradeoff between solution quality and running time is a critical issue. We propose two schemes: Intersections As Candidates (IAC) and Grids As Candidates (GAC) to find best candidate positions.

IAC provides a set of candidate RS positions including all the intersection points between any two SS’s feasible circles, which is illustrated in Fig.2(a). GAC provides a set candidate RS positions including all the center points of grids which divide the entire field. Fig.2(b) shows how GAC can be constructed. It is easy to see that the number of candidate positions is highly related to the grid size. Moreover, the smaller grid size, the more accurate results we can obtain. Thus, we set the grid size as small as possible as long as optimizer software (Gurobi 5.0) can find results.

![Fig. 2: Illustration of IAC and GAC](image-url)
reduce the power usage for the coverage RSs without changing coverage.

1) Coverage Under SNR Constraint: Given a relay network with a set of SS \( S = \{s_1, s_2, s_3, ..., s_n\} \) and SNR threshold \( \beta \). We first formulate an ILP with quadratic constraints \( \text{LPQC} \) to obtain optimal solutions. Let \( T_i \) and \( T_{ij} \) be the indicator variables in our \( \text{LPQC} \) , where \( T_i \) denotes if candidate position \( i \) is chosen to be placed RS in, and \( T_{ij} \) denotes if SS \( s_j \) has an access link with RS at position \( i \). The \( \text{LPQC} \) is listed as below:

Objective

\[
\min \sum_{\text{all } i} T_i \tag{3.1}
\]

Subject to :

\[
T_i \leq \sum_{\text{all } j} T_{ij} \leq nT_i \quad \forall i \tag{3.2}
\]

\[
\sum_{\text{all } j} T_{ij} = 1 \quad \forall j \tag{3.3}
\]

\[
d_{ij}T_{ij} \leq d_{ij}^* \quad \forall i \forall j \tag{3.4}
\]

\[
\sum_{\text{all } i} d_{ij}^*T_i - d_{ij}^*T_i \geq \beta T_{ij} \quad \forall j \tag{3.5}
\]

where (3.1) is the objective to find the minimum number of RS positions. Linear constraint (3.2) states that each placed RS covers at least one SS. Linear constraint (3.3) states that each SS can access to only one RS. Linear constraint (3.4) states feasible distance requirement for each SS. Quadratic constraint (3.5) states that each SS should satisfy its SNR constraint. Both IAC and GAC are used to generate the set of candidate positions. While the formulation will provide the minimum number of RSs that can provide feasible coverage to all SSs, and is used as benchmark for performance evaluation in later sections. The disadvantage of it is also obvious. With the number of SSs increasing, the running time of the formulation with quadratic constraints increases exponentially. Therefore we propose a polynomial-time solution as a practical solution for large networks, which is listed in Algorithm 1.

The first step, Algorithm Zone partition, is to partition the field into several zones such that SSs and RS in one zone will be distant from the stations in other zones. Thus, the interferences between inter-zone SS/RS pairs are small enough to be ignored. Zone partition algorithm is presented in Algorithm 2.

In Step 4, for each set of SSs, we first find a set of RSs to cover all the SSs satisfying distance requirements by solving a hitting set problem. [5] proposes a \( (1+\epsilon) \) approximation algorithm to solve minimum hitting set problem in geometry. Next, we aim to to satisfy the SNR requirements by massaging RS positions. We notice that if one SS is covered by only one RS, named one-on-one coverage, then this RS could be move closer to the covering SS (and hence further from other SSs).

In this way, it can save power for the SS and RS over access links, and reduce the possibility of interfering other SSs. Naturally, the more one-on-one coverage, the higher probability

Algorithm 1 SNR Aware Minimum Coverage (SAMC) \((S,D,\beta)\)

\begin{itemize}
  \item \text{Step 1} Initialize set \( L_{ss} = \{L_{ss}^1, L_{ss}^2, ..., L_{ss}^m\} \) which denotes SS groups to be returned from Zone Partition;
  \item \text{Step 2} \( L_{ss} \leftarrow \text{Zone Partition} (S,D) \);
  \item \text{Step 3} Initialize sets \( L_{RS} = \{L_{RS}^1, L_{RS}^2, ..., L_{RS}^{m}\} \) which denotes each coverage RS group placed for each SS group;
  \item \text{Step 4} for each SS group \( L_{ss}^i \) do
    \begin{itemize}
      \item \( K_{mhs} = \text{Minimum Hitting Set} (L_{ss}^i, D_s) \);
      \item \( G_i = \text{Coverage Link Escape} (L_{ss}^i, D_i, K_{mhs}) \);
      \item \( L_{RS} = \text{Sliding Movement} (G_i, L_{ss}^i, D_i, \beta) \);
    \end{itemize}
  \end{itemize}
  \item \text{Step 5} if any \( L_{RS}^i \) do return infeasible;
  \item \text{else} return \( L_{RS}^i \);
  \item \text{endfor}
\end{itemize}

Algorithm 2 Zone Partition Algorithm(S,D,N_{max})

\begin{itemize}
  \item \text{Step 1} calculate \( d_{max} \) according to \( N_{max} \),
    \begin{itemize}
      \item \( P_{max}Gd_{max} = N_{max} \), \( G = G_1+G_2h_2^2 \), \( N_{max} \) is the maximum noise which can be ignored;
    \end{itemize}
  \item \text{Step 2} create a new graph \( G \) involving all SSs in;
  \item \text{Step 3} for any two SSs \( s_i, s_j \) in \( G \) do
    \begin{itemize}
      \item \( d_{eff} = \min(\text{dist}(s_i, s_j)-d_j, \text{dist}(s_i, s_j)-d_j) \);
      \item \text{if} \( d_{eff} < d_{max} \) then
        \begin{itemize}
          \item add edge \( e(s_i, s_j) \) to \( G \);
        \end{itemize}
      \end{itemize}
  \end{itemize}
  \item \text{Step 4} find the connected components of \( G \);
  \item \text{Step 5} return SS groups of each connected component;
\end{itemize}

of satisfying SNR requirements for SSs. To seek more one-on-one coverage, Algorithm 3 Coverage Link Escape is used in Step 4 of Algorithm 1.

After Coverage Link Escape, it is still possible that some RSs which can provide feasible distance coverage but not SNR constraints for SSs. We call these place RS "infeasible RSs". So we present another scheme to reduce the number of infeasible RSs and improve the performance, which is listed in Algorithm 4. For each infeasible RS location, which is on each corresponding SS’s feasible circle, we try to “slide” the RS along the corresponding SS’s feasible circle to try to find a feasible RS location. The question is how to slide the infeasible candidates along SS’s feasible circles. The impact of sliding is complicated because it will not only affect the signal power received by its covering SSs but also the noise power received by other SSs. One SS may receive higher SNR at the cost of other SSs suffering lower SNR as the result of a sliding operation. Now the challenge is how to transfer the unlimited number of order combinations into limited ones so that we can solve it in polynomial time. One method is to find the infeasible coverage RSs which cannot satisfy SNR constraint. Based on the coverage topology, we try to slide the infeasible RSs along its covering feasible circles in order to clear SNR violations. If some SNR violation could not be cleared, then we mark its corresponding RS as un-slidable. After sliding all the infeasible RSs, we get a set of sliding
RSs and their updated locations. Since updating any one of
slidable RSs can change the coverage topology, every SS-
SNR constraint needs to be rechecked. We plan to try all the
possible combinations of updating RSs and then recheck all
the SS’s SNR constraints once each time of updating is done,
the procedures of which can be completed in polynomial time.
The details are in Algorithm 5. If all the SSs meet their SNR
requests, we found a feasible solution for the SAMC problem.
Otherwise, our algorithm will try remaining combinations.
The worse case is that all the updating combinations have
been tried but no one works. In this case, SAMC will return
infeasible.

In SAMC algorithm, we invoke minimum hitting set al-
gorithm to get the coverage RSs without considering SNR
constraint at beginning. Then we are checking if each SS’s
SNR could be met using coverage RSs topology. If there exist some SSs whose SNR constraints not satisfied, we
need to slide coverage RSs joint along its coverage SSs’
feasible circles in order to find a feasible solution. During
the proceeding of SAMC, no coverage RSs are deleted or
added in order to meet SSs’ SNR constraint. Consequently,
the result of SAMC has the same number of coverage RSs as
the number of using minimum hitting set algorithm. Therefore,
SAMC’s performance is highly related to minimum hitting set
algorithm, following the same scheme used in [1], [5]
gives an \((1 + \epsilon)\)-approximation PTAS for the minimum hitting
set problem. We adopted the PTAS, and claim that if SAMC
returns a feasible solution, it is also an \((1 + \epsilon)\)-approximation
solution. In other words, if SAMC returns a feasible solution
\(K\), the number of RS provide by \(K\) will be no more than

\begin{algorithm}
\caption{Coverage Link Escape(S,D,K_{\text{mhs}})}
\begin{algorithmic}
\State Step 1 construct a bipartite graph \(G\) between side \(A\) with all SSs, and side \(B\) including all the points in \(K_{\text{mhs}}\), where \(K_{\text{mhs}}\) is the RS set returned by minimum hitting set algorithm;
\State Step 2 for every SS \(s_i\) in side \(A\) do
\hspace{1em} for every point \(p_i\) in \(K_{\text{mhs}}\) do
\hspace{2em} if \(p_i\) is in or on \(c_i\), then
\hspace{3em} add edge \(e(s_i,p_i)\) to \(G\);
\hspace{1em} endfor
\hspace{0em} endfor
\State Step 3 calculate \(n_{\text{max}}\) ← the maximum number of edges including the same point in side \(B\);
\State Step 4 assume that all the edges in \(G\) and all the points in side \(B\) are not marked initially;
\State Step 5 for \(n\) from \(n_{\text{max}}\) to 1 do
\hspace{1em} for every unmarked point \(p\) in side \(B\) do
\hspace{2em} if there are \(n\) edges containing \(p\) then
\hspace{3em} mark these \(n\) edges;
\hspace{3em} mark point \(p\);
\hspace{3em} for each recent marked edge \(e(p,q)\) do
\hspace{4em} delete all the unmarked edges containing point \(q\);
\hspace{2em} endfor
\hspace{1em} endfor
\State Step 6 return bipartite graph \(G\);
\end{algorithmic}
\end{algorithm}

\begin{algorithm}
\caption{RS Sliding Movement(G, S, D, \(\beta\))}
\begin{algorithmic}
\State Step 1 \(H \leftarrow \emptyset, B \leftarrow \emptyset\);
\State Step 2 for every point \(p\) in side \(B\) of \(G\) do
\hspace{1em} if there is only one edge \(e(p,q)\) containing \(p\) then
\hspace{2em} if \(p\) and \(q\) are not at the same location then
\hspace{3em} move \(p\) to the same location as \(q\);
\hspace{2em} endif
\hspace{2em} \(H = H \cup \{p\}\);
\hspace{1em} delete point \(p\) and corresponding SS in \(G\);
\hspace{1em} endif
\State Step 3 for every SS \(s_i\) in side \(A\) do
\hspace{1em} check if SNR constraint \(\beta\) of \(s_i\) can be satisfied
\hspace{1em} if not, mark \(s_i\);
\State Step 4 \(B = B \cup \{\text{all marked } s_i\}\);
\State Step 5 if \(B\) is empty then
\hspace{1em} \(H = H \cup \{\text{all RSs in side } B\}\);
\hspace{1em} return \(H\);
\hspace{1em} else
\hspace{2em} \(R^*_a \leftarrow \text{all the RSs in side } B\) covering the SSs in \(B\);
\hspace{2em} \(R^*_b \leftarrow \text{all the rest RSs in side } B\);
\hspace{2em} \(H' \leftarrow \text{update RS topology } (R^*_a, R^*_b, G, S, D, H, B)\);
\hspace{2em} if \(H == H'\) then
\hspace{3em} \(H' \leftarrow \emptyset\);
\hspace{3em} return \(H'\);
\hspace{2em} else
\hspace{3em} return \(H'\);
\hspace{2em} endif
\hspace{1em} endif
\end{algorithmic}
\end{algorithm}

\begin{algorithm}
\caption{Update RS Topology \((R^*_a, R^*_b, G, S, D, H, B)\)}
\begin{algorithmic}
\State Step 1 copy \(R^*_a\) to \(R^*_c\);
\State Step 2 for each RS \(r_i\) in \(R^*_a\) do
\hspace{1em} \(W \leftarrow \emptyset\);
\hspace{2em} let \(s_k\) and \(s_j\) denote SSs whose SNR can and cannot be
\hspace{2em} met under coverage of \(r_i\), respectively;
\hspace{2em} construct a virtual circle \(c'_j\) for each \(s_j\) to ensure
\hspace{2em} that \(s_k\) SNR can be met only if \(r_i\) moves into \(c'_j\);
\hspace{2em} \(W=W\cup\{\text{all virtual circles } c'_j\}\cup\{\text{all feasible circles } c_k\}\);\nonumber
\hspace{2em} if all the circles in \(W\) have common area then
\hspace{3em} update \(r_i\) into anyplace in the common area;
\hspace{2em} else
\hspace{3em} mark \(r_i\) as unupdatable in \(R^*_a\);
\hspace{2em} endif
\hspace{1em} endfor
\State Step 3 for any combination \(C_i\) of updatable RSs in \(R^*_a\) do
\hspace{1em} copy \(R^*_a\) to \(R^*_c\);
\hspace{2em} update \(C_i\) relevant RSs in \(R^*_a\) to its corresponding location in \(R^*_c\);
\hspace{2em} check every SS’s SNR constraint;
\hspace{2em} if all SNRs satisfied then
\hspace{3em} \(H = H \cup R^*_a \cup R^*_b\); break;
\hspace{2em} else
\hspace{3em} record the unsatisfied SSs into a new set \(B'\);
\hspace{3em} if size(\(B')\) < size(\(B\)) then
\hspace{4em} \(R^*_a \leftarrow \text{all the RSs in side } B\) covering the SSs in \(B'\);
\hspace{4em} \(R^*_b \leftarrow \text{all the rest RSs in side } B\);
\hspace{4em} Update RS Topology \((R^*_a, R^*_b, G, S, D, H, B')\);
\hspace{4em} endif
\hspace{2em} endfor
\State Step 4 return \(H\);
\end{algorithmic}
\end{algorithm}
(1 + \epsilon) \cdot |OPT_C|$, where $OPT_C$ is an optimal solution with the minimum number of RSs that can provide feasible coverage.

2) **Power Reduction Optimization:** In previous section, we find feasible coverage RSs assuming that they are transmitting at their maximum power in SAMC. In this section, we aim to **adjust transmission powers** of the placed RSs so that we can further reduce the energy consumption while maintain coverage and SNR constraint.

Given a fixed network topology consisting of SSs and coverage RSs found by SAMC, we first present another Linear Programming with Quadratic Constraints (LPQC) to get an optimal RS transmission power allocation so that the total transmission power is minimized. Let $P_i$ denote the transmission power of $i$th RS in coverage SSs, which is in the range of $[0, P_{\text{max}}]$. For minimum number of RSs that can provide feasible coverage, $G$ denotes the coverage RS set as a result of SAMC, $P_{\text{min}}$ be the SNR threshold, $P_{\text{max}}$ be the maximum transmission power of RS, and $P_{ss}$ be the set of minimum received power requested by each SS to ensure its data rate. Moreover, let $P^{i}_{\text{snr}}$ denote the SNR power $P_{ss}$, for RS $r_i$, respectively. It is straightforward to calculate coverage power $P_c$ and SNR power $P_{\text{snr}}$ for each RS $r_i$. If all the RSs can reduce power to their own coverage power $P_c$, while meeting SNR constraint, the power saving approach is optimal. The details of the power saving algorithms are listed in **Algorithm 6**.

**Algorithm 6** Power Reduction Optimization (PRO) $(L_{\text{low}}, \text{S}, P_{ss}, \beta, P_{\text{max}})$

Step 1 $K \leftarrow \emptyset, P_1 \leftarrow \emptyset, P_2 \leftarrow \emptyset, P_3 \leftarrow \emptyset, P_{\text{temp}} \leftarrow \emptyset$.
Step 2 Initialize $P_1, P_2, P_3, P_{\text{temp}}$.
Step 3 for each item $i$ in $L_{\text{low}}$ do
   $P_1 = P_{\text{max}}; P_2 = P_{\text{max}}; \text{compute } P^{i}_{\text{min}}$;
   $P_2 = P^{i}_{\text{min}}; P_{\text{temp}} = P_{\text{max}}$;
   endfor
Step 4 put each RS point of $L_{\text{low}}$ into $K$;
Step 5 while ($K$ is not empty) do
   for each item $i$ in $P_1$ do
      if $(P_1 = P_2)$
         $P_1 = P_{\text{min}}$;
      endif
   endfor
   $P_1 = P_{\text{max}}$;
   $P_2 = P_{\text{snr}} - P_{\text{min}}$;
Step 6 $P_1 = P_{\text{snr}}$; $P_{\text{temp}} = P^{i}_{\text{snr}}$;
Step 7 remove RS point $i$ from $K$; $P_{\text{temp}} = P^{i}_{\text{snr}}$;
Step 8 $P^{i}_{\text{snr}} = P_{\text{snr}}$;
Step 9 clear $P_1$; $P_1 \leftarrow P_{\text{temp}}$;
   if length of $K$ is not changed for each item $i$ in $P_1$ do
      if $(P_1 = P_2)$
         compute $P^{i}_{\text{snr}}$;
      endif
   endif
Step 10 $P_1 = P_{\text{snr}}$; $P_{\text{temp}} = P^{i}_{\text{snr}}$;
Step 11 find index $i$ for minimum $\Delta P_1 = P^{i}_{\text{snr}} - P^{i}_{\text{min}}$;
Step 12 $P_1 = P_{\text{snr}}$; $P_{\text{temp}} = P^{i}_{\text{snr}}$;
Step 13 remove RS point $i$ from $K$;
endwhile
Step 14 return $\sum_{\text{all SS}} P^{i}_{\text{snr}}$;

**Theorem 1.** Algorithm 6 is a $(1 + \phi)$-approximation for the Power Reduction Optimization (PRO) problem. More specifically, if the power cost of all the RSs returned by Algorithm 6 is denoted by $|P|$, we have $|P| \leq (1 + \phi)|OPT_P|$, where $|OPT_P|$ is an optimal solution for PRO, and $\phi = \frac{\sum_i (P^{i}_{\text{snr}} - P^{i}_{\text{min}})}{OPT_P}$.

**Proof:** If all $P_{\text{snr}} \leq P_c$, then $|P| = |OPT_P|$. Otherwise, let $P_c$ denote the coverage power for RS $r_i$, and $P^{i}_{\text{snr}}$ denote the SNR power for RS $r_i$. Thus in whatever $OPT_P$ or $P$, it is composed of $P_c$ or $P^{i}_{\text{snr}}$ for each RS $r_i$. For instance,

$$P = \{P_c, P^{i}_{\text{snr}}, P^{i}_{\text{snr}}, P^{i}_{\text{snr}}, P_c\}$$

$$OPT_P = \{P_c, P^{i}_{\text{snr}}, P^{i}_{\text{snr}}, P^{i}_{\text{snr}}, P_c\}$$

Also, we let $I = \max_i \{\text{all } P^{i}_{\text{snr}} \text{ occur in } OPT_P\}$ and $C$ be the set of $i$ for all $i \in [1, I]$ in $OPT_P$ which does not operate in $P^{i}_{\text{snr}}$.
Therefore, the worse case for \( P \) is,
\[
P = OPT_P + \sum_{i \in C} (P_{i}^{snr} - P_{i}^{c})
\]

The approximation ratio in worse case is
\[
\frac{P}{OPT_P} = \frac{OPT_P + \sum_{i \in C} (P_{i}^{snr} - P_{i}^{c})}{OPT_P} = 1 + \frac{\sum_{i \in C} (P_{i}^{snr} - P_{i}^{c})}{OPT_P}
\]

Since \( \phi = \frac{\sum_{i \in C} (P_{i}^{snr} - P_{i}^{c})}{OPT_P} \), we have \( |P| \leq (1 + \phi) \cdot |OPT_P| \).

**B. Upper-tier Connectivity Relay Allocation (UCRA) problem**

On the upper tier, we need to consider how to transmit all the traffic from coverage RSs to BSs. We name the RSs placed on the upper tier connectivity RSs since the function of RSs in UCRA is to relay the communications between BS and coverage RSs. Similar to LCRA problem, we first assume that RSs relay with maximum power so that we can determine the minimum number of RS locations. In the second step, power optimization scheme will be applied to reduce the power consumptions.

**Definition 5** (Upper-tier Connectivity Relay Allocation (UCRA) problem). Given a wireless relay network with a set of coverage RSs \( R_c = \{r_1, r_2, ..., r_n\} \), distance requirements of \( R_c \) \( D_r = \{d_{r_1}^1, d_{r_2}^2, ..., d_{r_n}^p\} \), a set of base stations \( B = \{b_{s1}, b_{s2}, ..., b_{sn}\} \), UCRA seeks a minimum number of connectivity RSs operating with minimum power that ensures the communications between coverage RSs and BSs.

In [1], the authors studied a similar MUST problem, which is estimated to be NP-hard. MUST assumes only one BS and RSs always operate with maximum power. Therefore, MUST can be regarded as a special case of UCRA. To solve UCRA, the first challenge is how to decide the feasible distance of each RS, which is affected by the SSs or RSs being covered. In order to guarantee the data rate of each SS, for each RS \( r_i \), the link capacity between \( r_i \) and its parent node cannot be lower than the one between \( r_i \) and its any child. Therefore, we define feasible distance of an connectivity RS \( r_i \) connecting \( r_i \) and its parent station (an RS or a BS), should equals to the minimum feasible distance of all its children. With the assumption of connectivity RSs operating with \( P_{max} \), we propose our solution in **Algorithm 7**.

Since both MBMC and MUST proposed in [1] are minimum spanning tree based algorithms, MBMC has the same \( \frac{sd_{max}}{d_{min}} \)-approximation ratio as MUST, where \( d_{min} \) and \( d_{max} \) denote the minimum and maximum feasible distances of SSs, respectively. Having locations of connectivity RSs returned by MBMC, we then try to optimize power cost of each RS. Our solution is listed in **Algorithm 8**. Let \( L_{low} \) denote the set of coverage RSs, \( L_{high} \) denote the set of connectivity RSs, \( P_{ss} \) denote the set of received power requirements of SSs covered by RS \( r_i \), \( P_{rs} \) denote received power requirement of RS \( r_i \), \( N_i \) denote the number of RSs placed on the path from RS \( r_i \) to its parent, \( p_{ij} \) denote the transmission power of \( j^{th} \) RS on the path from RS \( r_i \) to its parent, and \( G = G_{i}G_{h}h_{1}^{2}h_{2}^{2} \).

**Algorithm 7** Multiple Base station Minimum Connectivity (MBMC) \((R_c, S, D, B)\)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>construct a complete graph ( G = (R_c, E) ), where ( R_c ) denotes coverage RS set;</td>
</tr>
<tr>
<td>2</td>
<td>( d_{min} = min_{i \in S} d_i );</td>
</tr>
<tr>
<td>3</td>
<td>for each node ( r_i ) in ( G ) do</td>
</tr>
<tr>
<td></td>
<td>create a new set ( K_i );</td>
</tr>
<tr>
<td></td>
<td>for each BS ( b_j ) in ( B ) do</td>
</tr>
<tr>
<td></td>
<td>calculate distance ( (r_i, b_j) ) and store it into ( K_i );</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
<tr>
<td></td>
<td>find min ( K_i ) and add the corresponding BS node ( b_i ) into ( G );</td>
</tr>
<tr>
<td></td>
<td>add edge ( e(r_i, b) ) into ( G );</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
<tr>
<td>4</td>
<td>for each edge ( e(x_i, x_j) ) in ( G ) do</td>
</tr>
<tr>
<td></td>
<td>assign weight ( w_1(x_i, x_j) = \left\lceil \frac{1}{d_{min}} \right\rceil - 1 ) on the edge;</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
<tr>
<td>5</td>
<td>Find a minimum spanning tree ( \tau_{mst} ) of ( G ) with BS as the root;</td>
</tr>
<tr>
<td>6</td>
<td>for each RS ( r_i ) do</td>
</tr>
<tr>
<td></td>
<td>Calculate its feasible distance ( d_i );</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
<tr>
<td>7</td>
<td>for each RS ( r_i ) and its parent ( r_p ) on ( \tau_{mst} ) do</td>
</tr>
<tr>
<td></td>
<td>( w_2(r_p, r_i) = 1 \left\lceil \frac{d_i}{d_{min}} \right\rceil );</td>
</tr>
<tr>
<td></td>
<td>Place ( w_2(z_p, r_i) ) RSs on ( e(r_p, r_i) ) separating the edge into [ \frac{d_i}{d_{min}} ] sections with each one with feasible distance;</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
</tbody>
</table>

| to its parent, \( p_{ij} \) denote the transmission power of \( j^{th} \) RS on the path from RS \( r_i \) to its parent, and \( G = G_{i}G_{h}h_{1}^{2}h_{2}^{2} \). |

**Algorithm 8** Upper-tier Connectivity Power Optimization (UCPO) \((L_{low}, L_{high}, P_{ss})\)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>for each RS ( r_i ) in ( L_{low} ) do</td>
</tr>
<tr>
<td></td>
<td>put each ( P_{ss} ) into new set ( K_i );</td>
</tr>
<tr>
<td></td>
<td>( P_{rs} = max(K_i) );</td>
</tr>
<tr>
<td></td>
<td>( D_i = \frac{\text{distance}(\text{parent}(i), i))}{N_i} );</td>
</tr>
<tr>
<td></td>
<td>( \phi = \frac{P_{rs}}{GD} );</td>
</tr>
<tr>
<td></td>
<td>for each RS ( r_j ) on path ( (i, \text{parent}(i)) ) do</td>
</tr>
<tr>
<td></td>
<td>( p_{ij} = \phi );</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
<tr>
<td></td>
<td>endfor</td>
</tr>
<tr>
<td>2</td>
<td>return ( \sum_{i \in S} \sum_{j \in \text{parent}(i)} p_{ij} );</td>
</tr>
</tbody>
</table>

**C. Approximation Algorithm for SAG problem**

With the approximation solutions (in terms of number of RSs placed) for both lower tier and upper tier, we present an approximation algorithm for the SAG problem in **Algorithm 9**.

**IV. NUMERICAL RESULTS**

In this section, numerical results are presented to show the effectiveness of our schemes, including SAMC, PRO, MBMC, UCPO and SAG algorithms. All the simulations are run on an Intel Core(TM)2 Duo CPU of 2.53GHz with 2GB types of memory. All the SSs and BSs are uniformly distributed in a square testing field. All the figures illustrate the average of 10 test runs for various scenarios.
Algorithm 9 SNR-aware Green (SAG) Relay $(S, D, B, \beta, P_{ss}, P_{max})$

Step 1 \( L_{low} \leftarrow \emptyset; L_{high} \leftarrow \emptyset; \)
Step 2 \( L_{low} \leftarrow \text{SAMC}(S, D, \beta); \)
Step 3 \( P_L \leftarrow \text{PRO}(L_{low}, S, P_{ss}, \beta, P_{max}); \)
Step 4 \( L_{high} \leftarrow \text{MBMC}(L_{low}, S, D, B, \beta); \)
Step 5 \( P_H \leftarrow \text{UCPO}(L_{low}, L_{high}, P_{ss}); \)
Step 6 \( P_{total} = P_L + P_H; \)
Step 7 return $P_{total}$;

A. Simulation Environment Settings

Since solving the ILP with quadratic constraints in Gurobi 5.0 [6] takes exponentially increasing running time and memory as growing the number of SSs or lessening the grid size, very large scale of testing field and huge number of SSs are not considered in our simulations. Considering the large scale of playing field is composed of a couple of small fields and the operations in each sub-field are independent to others. More specially, the entire testing field can divided into several sub-zones depending on the distributions of SSs in Zone Partition Algorithm since the influence of SSs or RSs from one sub-zone on the other sub-zones can be ignored. We select three scales of testing field to conduct the simulations. They are $300 \times 300$, $500 \times 500$, and $800 \times 800$, respectively. And we set the grid size as small as possible to avoid out-of-memory issue in the scenarios with consideration of the fact that the smaller the grid size is set, the more memory and more running time is taken. Signal-to-Noise Ratio (SNR) threshold for each SS is set in a range of [-10dB, -25dB]. Distance requirements of each SS is randomly distributed in [30,40]. The number of SSs in playing field varies from 5 to 70 at most, which are uniformly distributed. Now, we have five metrics to be compared among various scenarios such as the number of coverage RSs, power consumption of coverage RSs, the number of connectivity RSs, power consumption of connectivity RSs and the entire power consumption of all relay nodes. We present the numerical results in terms of separation of tiers. The results collected from lower tier are presented in IV-B and upper tier in IV-C, respectively.

B. Evaluation of Heuristics on Lower Tier

On the lower tier, we test the performance of IAC, GAC and SAMC on two playing fields of $500 \times 500$ and $800 \times 800$ respectively. The results are shown in Fig.3(a) to Fig.3(e). We can easily see that the number of coverage RSs coming from SAMC is lower than both GAC and IAC in whichever scenario. GAC has the most number of coverage RSs, which is reasoned by the selected size of candidate grid. The less size of candidate grid is set, the more accurate the results it would provide. Due to limited amount of memory in our simulation computer, we are not able to choose the small enough grid size so that we cannot get the accurate enough results from GAC. One more advantage of SAMC over both IAC and GAC is
that when the number of SSs grows above 50 in the $800 \times 800$ playing field and SNR is kept on $-15dB$, no feasible solutions are provided by both IAC and GAC but SAMC can stably find near-to-optimal solutions. In order to show both IAC and GAC can still work in the failed scenario, we decrease the SNR threshold to $-40dB$ and then IAC, GAC and SAMC all work, which is presented in Fig.3(c). From above analysis, we see whether IAC and GAC work correctly mainly depends on the SNR threshold. Thus we vary the SNR threshold from $-14dB$ to $-10dB$ and compare the performance between IAC and GAC, which is shown in Fig.3(d). From Fig.3(d), we find when SNR is increased to $-12dB$, IAC returns infeasible model but GAC returns 18 coverage RSs. Thus IAC is more sensitive to SNR than GAC. If grid size is decreased, GAC can provide more accurate results and has higher probability to get feasible solutions. From Fig.3(e) we find that GAC can still return feasible result with small grid size even under high SNR constraint ($-11.55dB$). In addition, let us look at the running time taken from IAC, GAC and SAMC on various testing field in Fig.4(b) and Fig.5(b). We can easily see that SAMC runs fastest among all the three and then IAC goes second. GAC takes a really long time to be completed, which is not good in practical usage. From the above discussion, we can say that SAMC outperforms both IAC and GAC from accuracy and timing aspects.

Fig.4(a) and Fig.5(a) show that PRO performs very near to optimal and it does save a large amount of power compared with baseline in which all the RSs operate in maximum power. Moreover, PRO can save more power especially in larger scale of testing field compared with baseline. Thus this result confirms our theoretical analysis of PRO performance.

**C. Evaluation of Heuristics on Upper Tier**

On the upper tier, we are concentrating on showing how MBMC works and why it outperforms MUST proposed in [1]. As we discussed in previous section, MUST can only be applied to one base station scenario but MBMC extends the function of MUST and works well in multiple base station environment, which is the common case in practical allocation. Thus we say that MBMC is more practical than MUST. Assume that 4 base stations are deployed in the testing field. We run MUST for four times, for each of which we let MUST connect to one of the four base stations, respectively. Fig.6(d) illustrates the case in which all SSs only connect to the corner BS (MUST algorithm) and Fig.6(c) illustrates the case in which all SSs connect to their nearest BS (MBMC algorithm). We can compare the data collected in Fig.4(c) and Fig.5(c). Apparently, MBMC outperforms MUST from any one of the scenarios adapting MUST. Also, we test the scenarios of various number of BSs from 1 to 4 on $500 \times 500$ playing field. From the Table II, we can easily find that the number of connectivity RSs from MBMC which need to be deployed is less than that from MUST in any testing scenarios. Based on the connectivity topology returned by MBMC, we confirm the performance of optimal UCPO in Fig.4(d) and Fig.5(d).

<table>
<thead>
<tr>
<th>BS</th>
<th>MUST BS1</th>
<th>MUST BS2</th>
<th>MUST BS3</th>
<th>MUST BS4</th>
<th>MBMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>41</td>
<td>45</td>
<td>N/A</td>
<td>N/A</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>45</td>
<td>45</td>
<td>N/A</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>47</td>
<td>46</td>
<td>45</td>
<td>45</td>
<td>40</td>
</tr>
</tbody>
</table>

**TABLE II:** Compare the performance between MBMC and MUST with various number of BSs in $500 \times 500$ field ($N_{SS} = 30$, SNR = $-15dB$)
D. Evaluation of Heuristics for SAG

Our SAG scheme combines all the solutions coming from both lower tier and upper tier. Fig.6(a), Fig.6(b) and Fig.6(c) illustrate the tree topologies coming from IAC plus MBMC, GAC plus MBMC and SAMC plus MBMC, respectively. At last, we compare the performance among SAG, SAMC+DARP, IAC+DARP and GAC+DARP, where DARP represents the deployment approaches discussed in [1] except the lower tier coverage approach since [1] does not take SNR constraint into account. Fig.7(a), Fig.7(b) and Fig.7(c) confirm that our design SAG is a great relay deployment strategy from the green aspect. Moreover, SAG is a more general case than DARP in practical deployment.

V. CONCLUSION

In this work, we studied the SNR-Aware Green (SAG) relay problem, which seeks the multi-hop relay node green allocation with channel capacity and SNR constraints in wireless relay networks. This problem is divided into two sub-problems, Lower-tier Coverage Relay Allocation (LCRA) problem and Upper-tier Connectivity Relay Allocation (UCRA) problem. For LCRA problem, we provided two approximation algorithms, SAMC and PRO, to solve the problem in two phases. Similarly, for UCRA problem, we proposed minimum spanning tree based approximation algorithm MBMC and optimal power optimization algorithm UCPO. At last, we combined the solutions of the LCRA and UCRA and presented a solution framework of SAG. Extensive numerical results have been conducted to support our theoretically analysis and show the good performances of our solutions.

REFERENCES