UAV-Mounted Mobile Base Station Placement via Sparse Recovery

MIN HUANG (Member, IEEE), LEI HUANG (Senior Member, IEEE), SHIDA ZHONG (Member, IEEE), AND PEICHANG ZHANG (Member, IEEE)
Guangdong Laboratory of Artificial-Intelligence and Cyber-Economics (SZ), Shenzhen University, Shenzhen 518060, China

Corresponding author: Lei Huang (lhuang@szu.edu.cn)

This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 61925108, in part by the Natural Science Foundation of China under Grant U1713217, Grant 61601304, Grant 61801302, Grant 61801297, and Grant 61801302, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515110509, in part by the Foundation of Guangdong Key Areas of “Service for Rural Revitalization Plan” under Grant 2019KZDZX2014, in part by the Founding of Shenzhen under Grant JCYJ20170302142545828 and Grant JCYJ20190808122005605, and in part by the Shenzhen University under Grant 2019119.

ABSTRACT

In order to deploy minimum number of unmanned aerial vehicle (UAV)-mounted mobile base stations (MBSs) to service all given ground terminals, this paper proposes an MBS placement based on sparse recovery (MBS-PBSR) algorithm. By exploiting the sparsity inherent in the differences between any two dedicated MBSs, the problem of UAV-mounted MBS placement could be formulated as an $\ell_0$-norm constrained optimization problem, which is then be solved by the reweighted $\ell_1$-norm method. Subsequently, the resulted solutions to the MBS placement are adjusted by the iterative redundant circle deletion algorithm, eventually leading to the redundant MBSs removal as much as possible. Simulation results demonstrate that our proposed MBS-PBSR algorithm works well with affordable computational complexity, and is nearly optimum in the sense of the number of deployed UAV-mounted MBSs.

INDEX TERMS

Unmanned aerial vehicle, mobile base station, sparse recovery.

I. INTRODUCTION

Due to its advantages of low cost, zero risk of casualties, strong survivability, good maneuverability and easy to use, unmanned aerial vehicle (UAV), also usually named as drone or remotely piloted aircraft, is more and more widely used in every corner of human society, such as reconnaissance, combat, planning, forest fire detection, communication, emergency search and rescue [1], [2].

Recently, since UAV-mounted mobile base stations (MBS) is able to provide flexible and rapid wireless connectivity, it is becoming more and more popular and thus attracting much research attention. For instance, when an emergency, such as an earthquake or power failure, occurs in a certain place, so that the terrestrial base stations cannot work normally, the UAV-mounted MBSs can be swiftly deployed to continue to provide wireless communication services for ground terminals (GTs) [3]–[5], as shown in Fig. 1. In addition to flexibility, UAV-mounted MBSs are cost-effective, compared with terrestrial base stations, in which the infrastructures, such as optical fibers, antenna towers and machine rooms, are expensive to purchase and install, especially in the places of battlefields or natural disaster. Furthermore, the altitude of UAV-mounted MBSs is generally much higher than that of terrestrial base stations, thus alleviating the interferences generated by mountains or buildings, and finally making their communication channels be dominated by line-of-sight (LoS) paths [6], [7].

However, since the transmitting power is constrained, the UAV-mounted MBS can only cover some GTs with limited distance. As a result, when a vast area needs to be provided emergency communication coverage, multiple UAV-mounted MBSs should be deployed at the same time.
Furthermore, in order to reduce the management and maintenance cost, the placement of the UAV-mounted MBSs should be optimized according to the known positions of all GTs, so as to minimize the number of MBSs.

In [8], the placement of a single UAV-mounted MBS is first formulated as an quadratically-constrained mixed integer nonlinear optimization problem, which is then be solved by the combination of the interiorpoint optimizer and bisection search. However, it only could provide communication service for a relatively small region. In order to extend its wireless coverage, the single UAV-mounted MBS is assumed to be mobile rather than static in [9]. Thus, the GTs could be served as long as they are under the flight route of the deployed UAVs. In fact, the idea behind [9] is similar to the time-division-multiple-access (TDMA) in cellular communications. Though all the GTs have been covered by the single mobile UAV-mounted MBS in terms of wireless communication, the throughput of each GT will significantly degrade, especially when the number of served GTs is relatively large. Therefore, in order to simultaneously meet the requirement of quality-of-service (QoS) for all GTs, we have to resort to deploying multiple UAV-mounted MBSs. In [10], the placement of multiple UAV-mounted MBSs could be considered as a Geometric Disk Cover (GDC) problem, where the radiiuses of all involved disks are fixed and same. Subsequently, the positions of deployed UAV-mounted MBSs could be determined by solving the so-obtained GDC problem through the Core-sets method, which, however, is indeed a global search method, leading to unaffordable computational complexity. In order to reduce the computational load, a strip-cover-with-disks algorithm is devised in [11] to simplify the GDC problem by decomposing it into two relatively simple subproblems. In addition, a K-means clustering algorithm is tailored in [12] to partition all the GTs to several clusters, each of which is served by its own MBS. More recently, a centralized deployment algorithm [14] and a centralized greedy search algorithm [15] are devised to deploy a small amount of UAVs for on-demand coverage while at the same time maintaining the connectivity among UAVs. Furthermore, a Spiral algorithm is proposed in [13], where UAV-mounted MBSs are sequentially deployed in an inward spiral manner until all GTs are served. However, all the above mentioned methods are heuristic algorithms, which may significantly degrade system performance, eventually increasing the number of UAV-mounted MBSs.

In order to further reduce the management cost, an MBS placement based on sparse recovery (MBS-PBSR) algorithm with moderate computational complexity is proposed in this paper. In particular, the problem of UAV-mounted MBS placement is first formulated as a sparse optimization problem, which is then be solved by the reweighted $\ell_1$-norm algorithm. Subsequently, the resulted solutions are adjusted by the iterative redundant circle deletion (IRCD) method so that the redundant MBSs are deleted as much as possible. Finally, numerical results show the superiority of our proposed MBS-PBSR algorithm in terms of the number of deployed UAV-mounted MBSs as well as computational complexity.

The rest of the paper is organized as follows. Section II introduces the problem of UAV-mounted MBS placement and illustrates its sparse representation. Our proposed MBS-PBSR algorithm is described in Section III. Simulation results are given in Section IV. Finally, conclusions are presented in Section V.

II. SPARSE REPRESENTATION

Assume $K$ GTs with indices of $\mathbb{K} = \{1, \cdots, K\}$ randomly distribute in a given area and their positions are known as $u_k = [u_{1,k}, u_{2,k}]^T \in \mathbb{R}^{2 \times 1}, k \in \mathbb{K}$, where $u_{1,k}$ and $u_{2,k}$, respectively, are the x and y coordinates of the $k$th GT on the ground plane. In addition, all the parameters, such as the transmitting powers and flying altitudes, of the UAV-mounted MBSs are considered as the same. Furthermore, the channels between GTs and MBSs are supposed to be dominated by LOS paths. Therefore, on the ground plane, the maximum communication radius of each MBS, denoted by $r$ could be regarded as the same.

In order to ensure communication validity for all $K$ GTs, each of them should locate within the communication radius $r$ of at least one MBS. On the other hand, in order to reduce the management cost as much as possible, we should pursue deploying minimum number of MBSs to service the given $K$ GTs, which, is the goal of this paper and could be formulated as [13]

\[
\begin{align*}
\text{(P1)} \quad & \min_{\{\vec{b}_m\}, m \in \mathbb{M}} |\mathbb{M}| \\
& \text{s.t.} \min_{m \in \mathbb{M}} ||u_k - \vec{b}_m||_2 \leq r, \quad \forall \ k \in \mathbb{K},
\end{align*}
\]

where $\mathbb{M} = \{1, \cdots, M\}$ stands for the indices of deployed MBSs, $M$ ($M \leq K$) represents the cardinality of $\mathbb{M}$, and $\vec{b}_m = [\vec{b}_{1,k}, \vec{b}_{2,k}]^T \in \mathbb{R}^{2 \times 1}$ is the position of the $m$th MBS.

From another point of view, each GT is assumed to be serviced by its own dedicated MBS. Thus, all the dedicated MBSs are sought to be deployed in as few positions as possible. Note that if some dedicated MBSs are simultaneously in the same position, only one MBS needs to be deployed in this position. Consequently, the problem of minimizing the number of deployed MBSs in P1 could be equivalent to that of minimizing the different positions of all dedicated MBSs, which could be rewritten in a sparse representation as

\[
\begin{align*}
\text{(P2)} \quad & \min_{\{\vec{b}_k\}, k \in \mathbb{K}} \sum_{i,j} I(b_i \neq b_j) \\
& \text{s.t.} \quad ||u_k - \vec{b}_k||_2 \leq r, \quad \forall \ k \in \mathbb{K},
\end{align*}
\]

where $\vec{b}_k = [\vec{b}_{1,k}, \vec{b}_{2,k}]^T \in \mathbb{R}^{2 \times 1}$ represents the position of the $k$th dedicated MBS which only services the $k$th GT, and $I(A)$ is an indicator function that returns 1 when the condition of $A$ is satisfied, otherwise 0.
III. MBS PLACEMENT BASED ON SPARSE RECOVERY ALGORITHM

A. INITIAL SPARSE OPTIMIZATION SOLUTIONS

For convenience, the position of GT or MBS is instead by a complex number which the real and imaginary components stand for the x and y coordinates, respectively. Mathematically, let \( u_k = u_{1,k} + iu_{2,k} \) and \( b_k = b_{1,k} + ib_{2,k} \), then the problem of P2 in (2) can be simplified as

\[
\text{(P3)} \quad \begin{cases} 
\min \left\{ \sum_{i,j} I(b_i \neq b_j) \right\} \\
\text{s.t. } \|u_k - b_k\|_2 \leq r, \forall \ k \in \mathbb{K}.
\end{cases}
\] (3)

In order to exploit the sparsity inherent in \( \sum_{i,j} I(b_i \neq b_j) \), the problem of P3 in (3) could be further equivalent to an \( \ell_0 \)-norm optimization problem [16, 17], which is given by

\[
\text{(P4)} \quad \begin{cases} 
\min \| \bar{p} \|_0 \\
\text{s.t. } D(u - b)^H D(u - b) \leq r I
\end{cases}
\] (4)

with

\[
\bar{p} = \text{vec} \left( \text{triu}(b^T \otimes 1 - b \otimes 1^T) \right) \in \mathbb{C}^{(k^2 - k) \times 1},
\] (5)

where \( b = [b_1, \ldots, b_K]^T \), \( u = [u_1, \ldots, u_K]^T \), \( \otimes \) denotes the Kronecker product, \( I \) represents a \( K \times 1 \) vector of all 1, \( \text{vec}[X] \), \( \text{triu}[X] \) and \( D[X] \) are the column-wise vectorization of \( X \), the \( (K - 1) \times (K - 1) \) upper triangular matrix extracted from \( X \) and the diagonal matrix created from vector \( x \), respectively. Note that vec (triu(\( b^T \otimes 1 - b \otimes 1^T \))) in (5) includes all the differences between any two elements of \( b \). Thus, \( \| \bar{p} \|_0 \) is the same as \( \sum_{i,j} I(b_i \neq b_j) \) in (3).

Furthermore, recalling that the positions of all GTs are assumed to be known, the distance between any two GTs could be computed by \( q = \text{vec} \left( \text{triu}(u^T \otimes 1 - u \otimes 1^T) \right) \). If the \( i \)-th element of \( q \), i.e., \( q_i \), is larger than \( 2r \), its corresponding dedicated MBSs are impossibility in the same position, indicating that \( \| \bar{p} \|_0 \) of P4 in (4) can be further simplified as

\[
\| \bar{p} \|_0 - d^H d + K^2/2 - K/2,
\] (6)

where the \( i \)-th element of \( d \) is represented as \( d_i = I(q_i) \leq 2r \) and \( p \) is such a vector that only keeps the \( i \)-th element in \( \bar{p} \) if \( d_i = 1 \). As a result, the problem of P4 in (4) can be further simplified as

\[
\text{(P5)} \quad \begin{cases} 
\min \| \bar{p} \|_0 \\
\text{s.t. } D(u - b)^H D(u - b) \leq r I
\end{cases}
\] (7)

However, the problem of P5 in (7) is an nonconvex and NP-hard problem, and thus requires an intractable search to reach its optimal solution, which is generally unattainable. In order to address this issue, the \( \ell_0 \)-norm optimization problem in (7) could be approximated as an \( \ell_1 \)-norm constrained optimization problem, which is convex. Nevertheless, the approximation usually makes the P5 fall into a local optimum. To alleviate this problem, the \( \ell_0 \)-norm in (7) could be replaced by the log-sum function [18], which could further reduce the approximation error as shown in Fig. 2. Therefore, we have

\[
\text{(P6)} \quad \begin{cases} 
\min \sum_i \log(|p_i| + \epsilon) \\
\text{s.t. } D(u - b)^H D(u - b) \leq r I
\end{cases}
\] (8)

where \( \epsilon \) is a constant used to control the tradeoff between the estimation accuracy and convergence speed. Though the problem of P6 in (8) is still nonconvex similar to \( \ell_0 \)-norm, it could be solved with the aid of the reweighted \( \ell_1 \)-norm algorithm proposed in [18]. The detailed procedure is described as follows.

1. Initialize the iteration count of \( t \) to 0 and the weight of \( W^{(t)} \) to \( D(1) \).
2. Estimate \( \hat{b} \) by solving the weighted \( \ell_1 \)-norm minimization problem, which is given by

\[
\hat{b}^{(t)} = \arg \min_b \| W^{(t)} p \|_1 \\
\text{s.t. } D(u - b)^H D(u - b) \leq r I.
\] (9)

3. According to (5) and (6), determine \( p^{(t)} \) along with the so-obtained \( \hat{b}^{(t)} \).
4. Update the weight of \( W \) by

\[
W_{i,i}^{(t+1)} = \frac{1}{|p_i^{(t)}| + \epsilon}.
\] (10)

5. If \( \|p^{(t)} - p^{(t-1)}\|_2^2 \) is small enough or \( t \) achieves the specified maximum number of iterations, stop the procedure. Otherwise, \( t = t + 1 \) and go back to step 2.

Finally, after convergence, we could get an initial sparse optimization solutions of \( \hat{b} = \mathcal{U}(b^{(1)}) \) for MBS placement, where \( \mathcal{U}(x) \) is an operator that only returns the same data in \( x \), but with no repetitions.

B. IRCD

Recall that all GTs randomly distribute in a given area and thus not all their dedicated MBSs are in the same position, resulting in nonzero elements existing in \( p \) of (9), some of which may be in a big difference. Consequently, even if the log-sum function instead of \( \ell_1 \)-norm is exploited to approximate \( \ell_0 \)-norm, the residual approximation error will still be large enough to degrade the estimation performance.
of $\hat{b}$ in (10). From the viewpoint of image processing, the GTs could be considered as points on the ground, and the coverage of MBSSs could be visualized as circles with centers $\hat{b}$ and radius $r$. The residual approximation error generated by the log-sum function will force these circles close to each other, thus separating some concentric circles, and eventually deploying more MBSSs. To combat this problem, we could employ the following IRCD algorithm, where the positions of circles are adjusted in previous iterations and then employed to further delete the redundant circles.

1) REDUNDANT CIRCLE DELETION

To begin with, the distance between the $i$th point and the $j$th circle could be calculated by

$$C_{i,j} = \left| \left( \hat{b}^T \otimes 1_{K \times 1} - u \otimes 1_{M \times 1}^T \right)_{i,j} \right|,$$

(11)

where $\hat{M}$ stands for the length of $\hat{b}$. Then, the $i$th point within the coverage of the $j$th circle could be determined by

$$E_{i,j} = I(C_{i,j} \leq r).$$

(12)

In particular, if the $i$th point could be served by the $j$th circle, $E_{i,j} = 1$. Otherwise, $E_{i,j} = 0$. Additionally, which point is covered by the $j$th circle could be identified by $e_j = [E_{1,j}, \ldots, E_{K,j}]^T$. Therefore, if all ones in $e_j$ could be found in the corresponding positions of $e_k$ ($k \neq j$), all the points covered by the $j$th circle are also covered by other circles, meaning that the circle from the $j$th MBSS is redundant and could be deleted. Mathematically, whether the $j$th circle could be removed depends on whether $f_j$ has elements greater than zero, where

$$f_j = e_j - \sum_{i \in \bar{\mathbb{D}}} e_i,$$

(13)

where $\bar{\mathbb{M}} = \{1, \ldots, \hat{M}\}$, $\mathbb{D}$ is a set that includes the indices of previous deleted circles, and $\setminus$ denotes set subtraction. After checking and deleting all the redundant circles, we should adjust the remaining circles, whose centers are recorded by $\hat{b}$.

2) CIRCLE ADJUSTMENT

First, define the matrices $C$ and $E$, whose elements in the $i$th row and the $j$th column have been, respectively, given by (11) and (12). Then, for $\hat{b}$, rather than $\hat{b}$, $C$ and $E$ are updated to $\tilde{C}$ and $\tilde{E}$, which are respectively obtained by deleting the $j$th columns of $C$ and $E$ with $j \in \mathbb{D}$. Subsequently, we could find that the $i$th point is covered by only one MBS, iff $i \in \bar{\mathbb{O}}$, where

$$\bar{\mathbb{O}} = \left\{ i | \sum_j \tilde{E}_{i,j} = 1 \right\}.$$  

(14)

In addition, we also observe that the $i$th point is on the edge of the circle from $\hat{b}$, iff $i \in \bar{\mathbb{F}}$, defined as

$$\bar{\mathbb{F}} = \left\{ i | \sum_j I(\tilde{C}_{i,j} = r) = 1 \right\}.$$  

(15)

Hence, the GTs with indices belonging to $\bar{\mathbb{G}}$ could be regarded as the points on the boundary of all circles, requiring us to cover them with as few circles as possible, where

$$\bar{\mathbb{G}} = \bar{\mathbb{O}} \cap \bar{\mathbb{F}}.$$  

(16)

Therefore, these circles generated from $\hat{b}$ need to be adjusted to cover points in $\bar{\mathbb{G}}$ as many as possible while including the existing points in $\bar{\mathbb{O}}$.

Specially, if the points lie within the $j$th circle, their indices should belong to $\mathbb{R} = \{i | \tilde{E}_{i,j} = 1\}$. Remembering (14), we could find the points with indices of $\mathbb{R}$ are only covered by the $j$th circle, where

$$\bar{\mathbb{R}} = \mathbb{R} \cap \bar{\mathbb{O}}.$$  

(17)

Then, the $j$th circle should be adjusted to cover more points in

$$\tilde{\mathbb{G}} = \bar{\mathbb{G}} - \mathbb{G} \cap (\bar{\mathbb{R}}),$$  

(18)

while covering the existing points in $\bar{\mathbb{R}}$. This could be implemented by the LocalCover algorithm proposed in [13]. Namely,

$$[\tau, \mathbb{H}] = \text{LocalCover}(\tilde{b}_j, \bar{\mathbb{R}}, \tilde{\mathbb{G}}).$$  

(19)

If $\mathbb{H}$ is not equal to $\tilde{\mathbb{G}}$, we adjust the center of the $j$th circle to $\tilde{b}_j = \tau$, update $\hat{b}$ to $\hat{b} = \tilde{b}$, and go back to Redundant Circle Deletion. Otherwise, we need to check the next circle until all the circles are evaluated. The detailed process of the proposed IRCD algorithm is tabulated in Algorithm 1.

**Algorithm 1 IRCD**

**Input:** Initial sparse optimization solutions $\hat{b}$;  
1) Redundant Circle Deletion  
1: According to (13), compute $f_j$ ($j \in \bar{\mathbb{M}}$) one by one in forward order, and collect the indices of the deleted circle in $\bar{\mathbb{D}}$ provided that there are elements in $f_j$ greater than zero;  
2: Remove all circles with indices in $\bar{\mathbb{D}}$, so that $\hat{b}$ is updated to $\tilde{b}$;  
2) Circle Adjustment  
3: Initialization: Set $j = 1$;  
4: Determine $\mathbb{R}$ and $\tilde{\mathbb{G}}$ by (17) and (18), respectively;  
5: Calculate $\tau$ and $\mathbb{H}$ by (19) based on the so-obtained $\tilde{b}_j$, $\mathbb{R}$ and $\tilde{\mathbb{G}}$;  
6: If $\mathbb{H} \neq \tilde{\mathbb{G}}$, adjust $\tilde{b}_j$ to $\tilde{b}_j = \tau$, update $\hat{b}$ to $\hat{b} = \tilde{b}$, and go back to 1. Otherwise, $j = j + 1$ and go back to step 4 until all circles are checked.  
**Output:** $\tilde{b}$.

Note that, as Subsection III-B is only composed by some simple logical operations, the computational complexity of our proposed MBS-PBSR algorithm mainly comes from Subsection III-A. Furthermore, it is easily observed that the solution of (9) dominates its computational burden in each iteration. Recalling (4) and (7), the number of elements in $W^{(t)}p$ of (9) could be readily computed by at most
\[ (9) \text{(will require)} \mathcal{O}(1/8K^6) \text{ complex operations according to [19]. Moreover, the convergence of the reweighted } \ell_1\text{-norm algorithm in Subsection III-A takes several iterations, which is generally no more than 10 [20]. Therefore, the total computational complexity of our proposed MBS-PBSR algorithm is not larger than } \mathcal{O}(5/4K^6), \text{ which is higher than that of the K-means [12] and Spiral [13] algorithms, but much less than that of the Core-sets approach [10] as shown in Table 1. However, it is almost the same as the Core-sets approach with few UAV-mounted MBSs to cover all GTs, as will be verified in next section.}

\textbf{IV. SIMULATION}

In this section, we evaluate the achievable performance of our proposed MBS-PBSR algorithm and also compare it with the K-means [12], Spiral [13] and Core-sets [10] algorithms. In the considered UAV-mounted MBS system, \( K \) GTs are randomly and independently scattered on the ground. In particular, both the \( x \) and \( y \) coordinates of any GT are assumed to follow the Gaussian distribution with mean zero and variance \( \rho \) km\(^2\). Furthermore, the maximum communication radius \( r \) projected on the ground by each MBS is supposed to be 0.5 km. In order to illustrate the advantage of our proposed MBS-PBSR algorithm intuitively and vividly, one final MBS deployment trial with \( K = 30 \) and \( \rho = 1 \) is

\begin{table}[h]
\centering
\begin{tabular}{ |c|c| }
\hline
Algorithm & Complex Operations \\
\hline
K-means & \( \mathcal{O}(\log(K)) \) \\
Spiral & \( \mathcal{O}(K^3) \) \\
MBS-PBSR & \( \mathcal{O}(5/4K^6) \) \\
Core-sets & \text{approx. } \mathcal{O}(M^2) \\
\hline
\end{tabular}
\caption{Computation complexity.}
\end{table}
shown in Fig. 3. From Fig. 3, we can observe that covering all GTs, the K-means algorithm needs 15 MBSs, the Spiral algorithm needs 13 MBSs, while our proposed MBS-PBSR algorithm only requires 12 MBSs, which is the same as the Core-sets approach does. As known to all, the performance of the K-means algorithm is vulnerable to the initializing centers and outliers, thus costing much more MBSs. Though the performance of the Spiral algorithm is better than that of the K-means algorithm, it deploys MBSs one by one along the convex hull, inevitably and easily falling into local optimum. By extensive search, the Core-sets algorithm only requires 12 MBSs. However, its running time is exponential in K as shown in the Table 1, leading to its computational complexity unaffordability, especially when K is large. On the contrary, our proposed MBS-PBSR algorithm starts from the global optimization assisted by sparse recovery, and then adjusts the approximate sparse optimization solutions by the iterative redundant circle deletion algorithm, so that it can cover all GTs with 12 MBSs only as the cost of moderate complexity.

Fig. 4 depicts the number of MBSs versus K for our proposed MBS-PBSR and the other three MBS placement algorithms. With the increase of K, the performance of the K-means algorithm becomes worse and worse, even unavailable in practice. This is because the goal of the K-means algorithm is not to minimize the number of MBSs, but to minimize the total Euclidean distances between each GT and its dedicated MBS. The Spiral algorithm is actually a heuristic search strategy, which deploys MBSs from outside to inside based on convex hull, difficulty forming an optimal MBS placement, as show in Fig. 3. In addition, since our proposed MBS-PBSR algorithm aims to minimize the number of MBSs based on the framework of sparse optimization, it could serve all GTs with fewer MBSs. This gap between our proposed MBS-PBSR and Spiral algorithms becomes more and more obvious as K increases. Although the performance of the Core-sets algorithm is slightly better than that of our proposed MBS-PBSR algorithm, it costs much more computational complexity.

Fig. 5 plots the number of MBSs versus ρ for different MBS placement algorithms. The increase of ρ means that all GTs are more dispersed, maybe resulting in the deployed MBSs increment. Even in this case, the performance of our proposed MBS-PBSR algorithm is still very close to that of the Core-sets algorithm as observed in Fig. 5. Furthermore, over the whole range of ρ, our proposed MBS-PBSR algorithm always outperforms the K-means and Spiral MBS placement approaches in terms of the number of deployed MBSs, especially when ρ is large.

V. CONCLUSION

A novel UAV-mounted MBS placement algorithm is proposed in the framework of sparse optimization, which is referred to as the MBS-PBSR algorithm. In our designed algorithm, the problem of UAV-mounted MBS placement is first formulated as a sparse optimization problem, then solved by the reweighted ℓ1-norm algorithm, and finally adjusted by the IRCD algorithm. Our results show that the proposed MBS-PBSR approach has significant performance advantage compared with the K-means and Spiral MBS placement algorithms in terms of the number of deployed UAV-mounted MBSs. Furthermore, compared with the optimal algorithm, namely the Core-sets algorithm, the performance of our proposed MBS-PBSR algorithm is slightly degraded, but it only costs moderate computational complexity.

REFERENCES

[1] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” IEEE Commun. Mag., vol. 54, no. 5, pp. 36–42, May 2016.
[2] A. F. B. Hanscom and M. A. Bedford, “Unmanned aircraft system (UAS) service demand 2015–2035 literature review & projections of future usage, version 0.1,” U.S. Dept. Transp., Washington, DC, USA, Tech. Rep. DOT-VNTSC-DoD-13-01, Sep. 2013.
[3] I. Bor-Yaliniz and H. Yanikomeroglu, “The new frontier in RAN heterogeneity: Multi-tier drone-cells,” IEEE Commun. Mag., vol. 54, no. 11, pp. 48–55, Nov. 2016.
MIN HUANG (Member, IEEE) received the B.Sc. degree in communication engineering and the Ph.D. degree in communication and information systems from Xidian University, China, in July 2008 and March 2015, respectively. From December 2016 to December 2017, he was a Visiting Scholar with the School of Electronics and Computer Science, University of Southampton, U.K. From June 2015 to August 2018, he was a Research Fellow with the College of Electrical and Computer Engineering, Shenzhen University, Shenzhen, China. He received the B.Sc. degree in communication engineering and the Ph.D. degree in communication and information systems from Xidian University, China, in July 2008 and March 2015, respectively. From December 2016 to December 2017, he was a Visiting Scholar with the School of Electronics and Computer Science, University of Southampton, U.K. From June 2015 to August 2018, he was a Research Fellow with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. He is currently an Associate Research Fellow in the area of wireless communications. His research interests include UAV-assisted communications, CFO estimation for OFDMA uplink systems, channel estimation, and OFDM signal detection.

PEICHANG ZHANG (Member, IEEE) received the B.Eng. degree (Hons.) in electronic engineering from the University of Central Lancashire, Preston, U.K., in 2009, and the M.Sc. and Ph.D. degrees in wireless communications from the University of Southampton, Southampton, U.K., in 2010 and 2015, respectively. He is currently with the College of Electronics and Information Engineering, Shenzhen University, China. His research interests include antenna selection, coherent and non-coherent detection, iterative detection, and channel estimation.