ORIGINAL RESEARCH PAPER

Interference coordination and throughput maximisation in an unmanned aerial vehicle-assisted cellular: User association and three-dimensional trajectory optimisation

Zhichao Yuan | Kaisheng Jiang | Weijie Jia | Rongwei Liu | Zhijun Wang | Xinyu Mao

Department of Electronics, Peking University, Beijing, China

Correspondence
Xinyu Mao, Department of Electronics, Peking University, Beijing 100871, China. Email: xymao@pku.edu.cn

Abstract
Unmanned aerial vehicles in wireless communications have caught great attention in unmanned aerial vehicle assisted communications and cellular-connected unmanned aerial vehicles. However, the unmanned aerial vehicle’s advantage of strong line-of-sight channels also brings about severe and extensive interference between unmanned aerial vehicle and the terrestrial communication networks and between multiple unmanned aerial vehicles. This paper jointly optimises the unmanned aerial vehicle’s three-dimensional trajectory and ground users’ association schemes with the unmanned aerial vehicle and ground base station to reduce interference. In consideration of fairness, the authors formulate the optimisation problem as maximising the minimum average throughput among all ground users, a mixed-integer non-convex problem that is challenging to tackle. To ease the difficulties, the authors have the optimisation problem relaxed and propose a suboptimal solution in the framework of the block successive upper-bound minimisation algorithm. Furthermore, an improved block successive upper-bound minimisation based penalty algorithm is obtained to avoid the local maximum point problem. Numerical simulations have validated our schemes.

1 | INTRODUCTION

Unmanned aerial vehicles (UAVs) have recently been found promising for wireless communications due to their advantages of strong line-of-sight (LOS) channels, high mobility, and flexible deployment [1, 2]. On the one hand, many types of UAV-assisted communications have been proposed, such as aerial base stations (BSs) [3], UAV-enabled relays [4, 5], multi-casting [6], and data collection [7]. On the other hand, UAVs can be used as aerial nodes to execute particular tasks while controlled by the terrestrial network called cellular-connected UAVs [8]. For example, a new kind of communication network named UAV-to-everything communications is put forward in depth [9, 10]. Besides, a UAV-enabled wireless power transfer system is studied in [11]. The UAVs could also be employed as intermediate aerial nodes between the macro and small cell tiers [12].

Moreover, one UAV could achieve multiple complex functions, as in [13], the UAV jointly optimises and performs sensing tasks and data transmission to the BS.

However, the strong LOS links also cause severe and extensive interference, leading to widespread performance degradation and design complication [14, 15]. Fortunately, a variety of researches have been carried out for interference coordination. For cellular-connected UAVs, the UAVs uplink cell associations and power allocations are jointly optimised in [14] to obtain optimal inter-cell interference coordination design and air-ground performance trade-off. A novel cooperative interference cancellation strategy by exploiting the existing backhaul connections is proposed in [16]. Moreover, a deep reinforcement learning algorithm is raised in [17] to design an interference-aware path planning scheme for a network of cellular-connected UAVs, which is further discussed in a cellular Internet of UAVs...
in [18, 19]. In [20], the author considers a cooperative cellular Internet that comprises multiple UAVs and has co-channel interference reduced through radio resource management and three-dimensional (3D) beamforming techniques. As for UAV-assisted communications, a dynamic mobility-aware interference avoidance scheme is proposed by utilising spectral graph theory [21]. The ground nodes’ transmit power allocation and scheduling and the UAV’s 3D trajectory design are jointly optimised with the existence of a group of ground jammers in [22]. A UAV-enabled interference channel is considered in [23], where a joint trajectory and power control problem is formulated to maximise the aggregate sum rate. In [24], directional antennas at the UAV and ground base station (GBS) are jointly applied to eliminate the mutual interference.

To mitigate co-channel interference with low costs, we propose a method of joint optimisation without consuming extra communication resources. The GBS and UAV user association schemes are jointly designed with the UAV’s trajectory, which enjoys a large degree of freedom. In this way, the UAV could always establish links with those ground users (GUs) less affected by the GBS. Furthermore, it would keep a distance from the GU that is communicating with the GBS at the same time. In respect of the GBS, it would switch to those GUs far away from the UAV in real time. With an appropriately designed algorithm, the targets of avoiding interference, improving throughput, and maintaining the fairness between users could be balanced and achieved in the meantime.

This paper investigates a UAV-assisted hybrid cellular network, where a UAV and a GBS are concurrently deployed to serve several GUs. The UAV and GBS utilise the same frequency band, so co-channel interference exists when they are simultaneously working. A time-division multiple access (TDMA) scheme is adopted. Having considered the fairness between GUs, a mixed-integer non-convex optimisation problem is formulated to maximise the minimum average throughput among all GUs under certain UAV trajectory and user association constraints. The UAV and GBS user association schemes, along with UAV’s trajectory, are jointly optimised to coordinate the interference and maximise the throughput. Intuitively, the GBS would choose those GUs farther from UAV to reduce UAV’s interference on its users, while the UAV would behave similarly. The formulated problem is quite challenging to solve, so we first have the integral constraints relaxed to be continuous. Then a suboptimal solution to the relaxed problem is proposed applying an algorithm in the framework of the block successive upper-bound minimisation (BSUM) algorithm [25]. Furthermore, an improved BSUM-based penalty algorithm is put forward to avoid the local maximum points problem. Our scheme and algorithms are validated to be effective by numerical simulations.

The rest of this paper is organised as follows. The background and principal work of our research are introduced in Section 1. In Section 2, we present the system model and formulate the optimisation problem to be solved. Two algorithms to tackle the formulated problem are proposed in Section 3, while the former is based on the BSUM algorithm, and the latter is improved with penalty factors. The convergence behaviours of these algorithms are also discussed. Numerical simulation results are provided in Section 4. Finally, this paper is concluded in Section 5.

2 | SYSTEM MODEL AND PROBLEM FORMULATIONS

2.1 | System model

As shown in Figure 1, we consider a UAV-assisted single-cell wireless communication network. In the system, a UAV and GBS are jointly deployed to serve a group of K GUs. The GUs are generated by the mixed traffic model in [26]. Two sets of GUs are generated via an independent spatial Poisson point process with different densities \( \lambda_u \) and \( \lambda_m \). Then GUs with density \( \lambda_u \) are shifted towards GBS by correlation parameter \( \rho \), meaning the distances between these GUs and the GBS are resized from \( R \) to \( R(1 - \rho) \). All GUs are indexed by the set \( \mathcal{K} = \{1, \ldots, K\} \). A 3D Cartesian coordinate system is defined, where the GBS is fixed at \( q_G = (x_G, y_G, h_G) \), and GU \( k \) is denoted by \( q_k = (x_k, y_k, 0) \). The UAV is assumed to fly periodically and its position at time instant \( t \) is \( q(t) = (x(t), y(t), z(t)) \). For ease of exposition, all the \( x \) and \( y \) coordinates are combined to form 2D vectors \( \mathbf{w} = (x, y) \).

To avoid ground obstacles and obey government regulations, the UAV’s altitude is restricted. The UAV’s horizontal and vertical velocities are assumed to be controlled independently, and they cannot exceed their maximum limits \( V_{\text{max}}^h \) and \( V_{\text{max}}^v \). Thus, we obtain the following constraints on UAV’s trajectory:

\[ q(0) = q(T), \]

\[ H_{\text{min}} \leq z(t) \leq H_{\text{max}}, \quad \forall \ 0 \leq t \leq T, \]

where \( H_{\text{min}} \) and \( H_{\text{max}} \) are the minimum and maximum altitudes, respectively.
\[ \| \mathbf{w}(t) \| \leq V_{\text{max}}^h, \quad \forall 0 \leq t \leq T, \]
\[ |z(\ell)| \leq V_{\text{max}}^v, \quad \forall 0 \leq \ell \leq T, \]

where \( T \) represents the flight period, and (1) is the periodicity requirement. \( T \) is equally discretised into \( N_t \) time slots for ease of design, where the element slot length \( \delta_t \), is chosen to be sufficiently small that the communication conditions can be considered approximately unchanged during it. Therefore, the UAV’s trajectory \( q(t) \) can be approximated by a sequence \( \{q[n]\} \), where \( q[n] = q((n-1)\delta_t) \) represents UAV’s position at time slot \( n \) \( \in \mathcal{N} = \{1, \ldots, N_t + 1\} \). Then the trajectory constraints (1)–(4) can be approximately written as
\[ q[1] = q[N_t + 1], \quad (5) \]
\[ H_{\min} \leq z[n] \leq H_{\max}, \quad \forall 1 \leq n \leq N_t + 1, \quad (6) \]
\[ \| w[n+1] - w[n] \| \leq V_{\text{max}}^h \delta_t, \quad \forall 1 \leq n \leq N_t, \quad (7) \]
\[ |z[n+1] - z[n] | \leq V_{\text{max}}^v \delta_t, \quad \forall 1 \leq n \leq N_t. \quad (8) \]

Only downlink communications from the GBS and UAV to GUs are considered. The GBS and UAV are assumed to be equipped with an omnidirectional antenna, and they transmit data with constant power \( P_G^k \) and \( P_U^k \).

To facilitate the problem, we assume there exists only large-scale fading for the GBS–GU links. The signal power that GU \( k \) receives from the GBS is modelled as
\[ P_{G_k}^k = P_G^k g_k u_k, \quad (9) \]
where \( g_k = \frac{g_0}{(d_k^G)^\alpha} \) accounts for path loss. \( g_0 \) denotes the reference channel power gain at \( d_0 = 1 \) m, while \( d_k^G = \| q_G - q_k \| \) denotes the distance between the GBS and GU \( k \), and \( u_k \) denotes the path loss exponent. \( u_k \) is an independent and identically distributed log-normal random variable describing the shadow fading. Specifically, the decibel of \( u_k \) follows a normal distribution with mean value 0, standard deviation \( \sigma \), that is, \( 10 \log_{10} u_k \sim \mathcal{N}(0, \sigma^2) \).

The UAV is assumed to communicate with GUs by LOS links, and Doppler effects caused by the UAV’s motion is assumed to be well compensated. Therefore, the signal power GU \( k \) receives from the UAV can be described as
\[ P_{U_k}^k = P_U^k b_k, \quad (10) \]
where \( b_k = \frac{b_0}{(d_k^U)^\beta} \) is the free-space path loss of the LOS channel, with \( b_0 \) denoting the channel power gain at the reference distance \( d_0 = 1 \) m. \( d_k^U[a] = \| q[a] - q_k \| \) denotes the distance between UAV and GU \( k \) at time slot \( n \).

The GBS and the UAV are assumed to utilise a common frequency spectrum with a bandwidth \( B \), and the TDMA scheme is adopted to serve all GUs. Binary variables \( a_k^G[n] \) and \( a_k^U[n] \) are defined to describe the association between the UAV, GBS, and GUs. \( a_k^G[n] = 1 \) indicates that the GU \( k \) is served by the GBS in time slot \( n \), and \( a_k^U[n] \) means similarly. We assume that the GBS and UAV can only serve one GU at a time separately. A GU cannot be associated with both the GBS and the UAV at the same time, too. Below are the constraints on user association schemes:
\[ \sum_{k=1}^K a_k^G[n] \leq 1, \quad \forall n, \quad (11) \]
\[ \sum_{k=1}^K a_k^U[n] \leq 1, \quad \forall n, \quad (12) \]
\[ a_k^G[n] + a_k^U[n] \leq 1, \quad \forall k, n, \quad (13) \]
\[ a_k^G[n] a_k^U[n] \in \{0, 1\}, \quad \forall k, n. \quad (14) \]

The GUs are assumed to receive additive white Gaussian noise with power spectral density of \( N_0 \). Taking co-channel interference into consideration, the signal-to-interference-plus-noise ratio (SINR) for GUs associated with the GBS is
\[ \gamma_k^G[a] = \frac{P_G^k g_k u_k}{P_U^k b_k[a] \sum_{k=1}^K a_k^U[n] + B N_0}, \quad (15) \]
SINR for GUs served by the UAV is
\[ \gamma_k^U[a] = \frac{P_U^k b_k[a]}{P_G^k g_k u_k \sum_{k=1}^K a_k^G[n] + B N_0}. \quad (16) \]
Therefore, we obtain the throughput of GU \( k \) in time slot \( n \):
\[ R_k[a] = a_k^G[n] B \log_2 \left( 1 + \frac{P_G^k g_k u_k}{P_U^k b_k[a] \sum_{k=1}^K a_k^U[n] + B N_0} \right) + a_k^U[n] B \log_2 \left( 1 + \frac{P_U^k b_k[a]}{P_G^k g_k u_k \sum_{k=1}^K a_k^G[n] + B N_0} \right). \quad (17) \]
\( a_k^G[n] \) and \( a_k^U[n] \) are coupled together in (17), making the user association problem difficult to tackle. Considering that there always exist users far apart from each other, the GBS and UAV would, respectively, choose these users to serve almost at all the time to achieve higher throughput. Therefore we make an
Our goal is to maximise
\[ a \] is hard to solve, so we relax the binary constraints on

\[ (P1) \] is a mixed-integer non-convex optimisation problem that
are assumed to be precisely known. For the sake of fairness,
position of GBS and GUs, as well as channel state informa-
tion, are assumed to be precisely known. For the sake of fairness,
with UAV and GBS, along with the UAV’s trajectory,
and UAV trajectory. In the user association problem, we
partition the entire optimisation variables into two blocks,
are coupled together. To facilitate this problem, we apply
the block coordinate descent method [25] and
partition the entire optimisation variables into two blocks, \( A = \{ a^G_k[a], a^U_k[a] \} \) and \( Q = \{ q[a] \} \). Then problem (P2) could be
decomposed into two optimisation problems of user association
and UAV trajectory. In the user association problem, we
hold \( Q \) fixed and optimise \( A \). In the UAV trajectory optimisation
problem, we hold \( A \) fixed and optimise \( Q \). Then, these two
subproblems needed to be solved alternatively to approaching
the optimisation solution if (P2).

However, the decomposed UAV trajectory problem is still
challenging as it is a non-convex optimisation problem. Besides,
\( R^U_k[a] \)'s logarithmic expression in \((19)\) would consume much
computation resources, leading to an increase in computing
time. Therefore, we introduce a locally tight lower bound to
substitute the objective and successively approximates the original
objective with a sequence of convex approximations (SCA)
[27].

We describe the whole optimisation algorithm with an effi-
cient algorithmic framework called the BSUM and call it ‘the
relaxed BSUM algorithm’. The details are given in Subsection
3.1. As BSUM is proposed for large-scale data optimisation,
our problem could be solved with high efficiency applying
this framework. Furthermore, a parallel algorithm called par-
allel successive convex approximation is also included in the
BSUM framework. Therefore, the BSUM makes it easier for
us to change our current algorithm into a parallel version, also
our next research target. Besides, our current algorithm faces
the problem of local maximum points. Specifically, the algo-
rithm may plunge into a certain point that is a local maximum
point but not the global maximum point, and output a solution
whose performance is much lower than the global optimisa-
tion value. But within the BSUM framework, we could consider
more flexible variable update rules rather than what we use in the
current algorithm. In fact, we are now in process of looking
for a suitable update rule to jump out of local maximum
points.

Apart from the method discussed above, we propose another
way to deal with the local maximum points problem in Sub-
section 3.2. An improved solution that introduces a penalty
factor and adjusts the iteration strategy is described in that
subsection. In Subsection 3.3, the connections between the
original problem (P1) and the transformed problems are
given, and these two algorithms’ convergence behaviours are
discussed.
3.1 | The relaxed BSUM algorithm

As has been stated, the optimisation problem (P2) is decomposed here into two subproblems: one to optimise the user association schemes assuming the UAV trajectory is known, and the other to optimise the UAV’s trajectory with given user association schemes.

3.1.1 | Subproblem of user association optimisation

With the binary constraint (14) relaxed and the UAV trajectory \( \{q[n]\} \) given, the user association question is as P3:

\[
P_3:\quad \max_{\substack{\alpha_k^G[n], \alpha_k^U[n], R_{\min} \in \mathbb{R}^N \cap \mathbb{N} \ni k \in K, a \in A}} R_{\min}
\]

\[
s.t.\quad \frac{1}{N_j} \sum_{n=1}^{N_j} \left[ \alpha_k^G[n] \log_2 \left( 1 + \frac{P_k^G g_k h_k}{\beta_k[n] + B N_0} \right) + \alpha_k^U[n] \log_2 \left( 1 + \frac{P_k^U b_k[n]}{P_k^G g_k h_k + B N_0} \right) \right] \geq R_{\min}, \quad \forall k, n
\]

\[
\alpha_k^G[n], \alpha_k^U[n] \in [0, 1], \quad \forall k, n
\]

(11) \sim (13).

This is a standard linear programming (LP) problem, which can be solved by today’s mathematical solvers with high efficiency.

3.1.2 | Subproblem of UAV trajectory optimisation

Assuming that the user association variables \( \alpha_k^G[n], \alpha_k^U[n] \) are given, the UAV’s trajectory optimisation problem can be denoted as

\[
P_4:\quad \max_{q[n], R_{\min} \in \mathbb{R}^N \cap \mathbb{N}} R_{\min}
\]

\[
s.t.\quad \frac{1}{N_j} \sum_{n=1}^{N_j} \left[ \alpha_k^G[n] \log_2 \left( 1 + \frac{P_k^G g_k h_k}{\beta_k[n] + B N_0} \right) \right] + \alpha_k^U[n] \log_2 \left( 1 + \frac{P_k^U b_k[n]}{P_k^G g_k h_k + B N_0} \right) \geq R_{\min}, \quad \forall k, n
\]

(5) \sim (8).

With the definition of some constants

\[
S_k^G = \frac{P_k^G g_k h_k}{B N_0}, \quad S_k^U = \frac{P_k^U b_k}{B N_0}, \quad S_0^U = \frac{P_k^U b_k}{P_k^G g_k h_k + B N_0}
\]

and expressions,

\[
E_k^G[a] = \log_2 \left( 1 + \frac{S_k^G \|q[n] - a\|^2}{S_k^U + \|q[n] - a\|^2} \right),
\]

\[
E_k^U[a] = \log_2 \left( 1 + \frac{S_0^U}{\|q[n] - a\|^2} \right),
\]

(25b) can be rewritten in a more compact form:

\[
\frac{1}{N_j} \sum_{n=1}^{N_j} \left[ (\alpha_k^G[n] E_k^G[a] + \alpha_k^U[n] E_k^U[a]) \right] \geq R_{\min}, \quad \forall k (26)
\]

Constraint (26) accounts for the main difficulties in solving problem P4, as \( E_k^G[a] \) and \( E_k^U[a] \) are both complex but neither convex nor concave functions of \( q[n] \). However, if we consider \( \|q[n] - a\|^2 \) as a variable, then \( E_k^G[a] \) is a concave function, and \( E_k^U[a] \) is a convex function. Thus problem P4 is structurally similar to the classic concave–convex problem in [28].

Like the procedures in solving concave–convex problems, we have sought a lower bound to approximate the convex part \( E_k^U[a] \). Considering the rule that a convex function is globally lower bounded by its tangent line, a lower bound of \( E_k^U[a] \) near local point \( \{q[n]\} \) is

\[
E_k^U[a] = \left. \frac{d E_k^U[a]}{d q[n]} \right|_{q[n] = a} (\|q[n] - a\|^2 - \|q[n] - a\|^2) + E_k^U[r][a],
\]

(27)

where \( E_k^U[a] \) is the derivative of \( E_k^U[a] \) to \( \|q[n] - a\|^2 \) at the local point \( \|q[n] - a\|^2 \), and \( E_k^U[r][a] \) is \( E_k^U[a] \)’s value. Constants \( \lambda_k^G[a], \lambda_k^U[a] \) can be given by

\[
\lambda_k^G[a] = \frac{- \log_2 [(S_0^U + \|q[n] - a\|^2) / (\|q[n] - a\|^2 + S_0^U)]}{\|q[n] - a\|^2},
\]

\[
\lambda_k^U[a] = \log_2 \left( 1 + \frac{S_0^U}{\|q[n] - a\|^2} \right),
\]

\( \hat{E}_k^U[a] \) is a quadratic function of \( q[n] \) and always concave as \( \lambda_k^U[a] \) is always negative. This is an attached benefit of our process of finding a lower concave bound of \( E_k^G[a] \) concerning \( \|q[n] - a\|^2 \).

Although \( E_k^G[a] \) is a concave function of \( \|q[n] - a\|^2 \), it is not concave concerning \( q[n] \). As \( E_k^G[a] \) monotonously increases with \( \|q[n] - a\|^2 \) increasing, and \( \|q[n] - a\|^2 \) is concave of \( q[n] \), we substitute \( \|q[n] - a\|^2 \) with its lower linear bound. In this method, \( E_k^G[a] \) could be estimated as a concave function of \( q[n] \).
The lower linear bound of \( \|q[n] - q_k\|^2 \) obtained at a local point \( \{q'[n]\} \) is represented as

\[
2(q'[n] - q_k) \cdot (q[n] - q'[n]) + \|q'[n] - q_k\|^2 \leq \|q[n] - q_k\|^2, \quad \forall \ k, n. \tag{28}
\]

For ease of statements, a slack variable \( d^k_{q[n]} \) is introduced in place of the lower linear bound of \( \|q[n] - q_k\|^2 \):

\[
d^k_{q[n]} \leq 2(q'[n] - q_k) \cdot (q[n] - q'[n]) + \|q'[n] - q_k\|^2, \quad \forall \ k, n. \tag{29}
\]

As the two sides of (29) are always equal in the optimal point, \( d^k_{q[n]} \) is an equivalent substitution of the lower linear bound shown above.

Then the lower bound of \( E^G_k[n] \) is given as

\[
E^{G_l}_k[n] = \log_2 \left( 1 + \frac{d^k_{q[n]} E^{G_l}_k[n]}{S^k_{E} + d^k_{\text{q}[n]}} \right) \leq E^G_k[n]. \tag{30}
\]

Till now, constraints (26) have been transformed into the standard convex constraints on arguments \( \{q[n], d^k_{q[n]}\} \):

\[
\frac{1}{N_i} \sum_{s=1}^{N_i} \left( d^k_{q[n]} E^{G_l}_k[n] + a^U_k[n] E^U_k[n] \right) \geq R_\min, \quad \forall \ k. \tag{31}
\]

However, in the existence of the logarithmic expression of \( E^G_k[n] \)'s lower bound, it is still high demanding to deal with the constraints (31). Therefore, we propose a method to approximate and calculate the logarithmic expression by replacing it with a concave fractional function:

**Remark 1.** For \( x \in [0, +\infty) \), function \( f(x) = \ln(1 + x) \) has a global lower-bounded approximation as

\[
g(x, x_0) = C(x_0) - \frac{1}{a(x_0)x + b(x_0)}. \tag{32}
\]

\( a(x_0), b(x_0), C(x_0) \) are coefficients calculated at local point \( x_0 \), and we have carefully chosen their values to make \( g(x, x_0) \) as close to \( f(x) \) as possible. See the Appendix for details.

Define

\[
x_k[n] = \frac{S^G_k d^k_{q[n]}}{S^k_{E} + d^k_{\text{q}[n]}}, \quad x_{0,k}[n] = \frac{S^G_k d^k_{q[n]}}{S^k_{E} + d^k_{\text{q}[n]}},
\]

then

\[
E^{G_l}_k[n] = \frac{1}{\ln 2} g(x_k[n], x_{0,k}[n]).
\]

The corresponding lower fractional bound can be expressed as

\[
E^{G'}_k[n] = \frac{1}{\ln 2} g(x_k[n], x_{0,k}[n])
\]

For simplicity, we denote \( a(x_0[n]) \cdot \ln 2, b(x_0[n]) \cdot \ln 2, C(x_0[n]) / \ln 2 \) as \( a, b, C \), and then \( E^{G'}_k[n] \) can be rewritten as

\[
E^{G'}_k[n] = C - \frac{1}{a s^{G'}_k + b} \left[ 1 + \frac{a s^{G'}_k s^{U}_k}{a s^{G'}_k + b} d^{U}_k[n] + d^{U}_k[n] \right]. \tag{33}
\]

It could be verified with a simple analysis that \( E^{G'}_k[n] \) is concave concerning \( d^{U}_k[n] \).

Finally, the subproblem of UAV trajectory optimisation is approximated as a standard convex optimisation problem which can be efficiently solved by standard convex optimisation solvers, in the form of the following:

\[
\text{P5:} \max_{q[n], R_\min, d^k_{q[n]}[n]} R_\min \tag{34a}
\]

s.t. \( \frac{1}{N_i} \sum_{s=1}^{N_i} [a^U_k[n] E^{G'}_k[n] + d^U_k[n] E^U_k[n]] \geq R_\min / B, \quad \forall \ k, \tag{34b} \)

\[
2(q'[n] - q_k) \cdot (q[n] - q'[n]) + \|q'[n] - q_k\|^2 \\
\geq d^k_{q[n]}, \forall \ k, n, \tag{34c} \)

(5) \sim (8).

### 3.1.3 Initialisation algorithm

In our algorithms, the UAV's trajectory is optimised in an iteration way that needs an initial trajectory. Besides, the user association problem is also optimised, assuming a known trajectory of the UAV. Thus, an initialisation scheme of the UAV’s trajectory is required as the beginning of our two algorithms.

For simplification, we divide all GUs into two categories by power \( P^G_{E,k} \) of the signal they received from the GBS. Those GUs receiving higher \( P^G_{E,k} \) than \( P^G_0 \) are initially associated with the GBS while the others are linked to the UAV. Denote the indexes of UAV’s users as \( K^U = \{K^U_1, \ldots, K^U_{K^U}\} \), where \( K^U \) is the total number of UAV’s users. Temporarily assuming that the UAV’s altitude is fixed at \( H_0 \), then the UAV and its associated GUs could be extracted as a simpler network: a UAV serving multiple fixed ground users. This simpler network has been discussed comprehensively in previous work like [29, 30].

Intuitively, we consider two extreme cases concerning the UAV’s flight period \( T \):

- (1) When \( T \to 0 \), the UAV has to stay at one point \( q^U_0 = (x_{0,0}, y_{0,0}, H_0) \) all the time. Best values of \( x_{0,0}, y_{0,0} \) could be obtained
by solving the following problem:

\[
P_6: \quad \max_{\mathbf{x}_i, \mathbf{j}} \mathcal{R}_{\min}^U \quad \text{s.t.} \quad \log_2 \left[ 1 + \frac{P_t^U b_0}{(P_t^U g_{x_i}^U + \mathcal{B}N_i)\|\mathbf{q}_i^U - \mathbf{q}_0^U\|^2} \right] \geq \frac{\mathcal{R}_{\min}^U}{B}, \quad 1 \leq i \leq K_U.
\]

This problem aims to maximise the minimum communication rate \(\mathcal{R}_{\min}^U\) among all UAVs’ users by optimising \(x_i, j_i\). Problem \(P_6\) is also non-convex, and we handle it with an iterative method that we have introduced in the trajectory optimisation subproblem. Limited by space, we omit the details to solve problem \(P_6\) and only show the initial point of iterations here.

\[
x_c = \frac{1}{K_U} \sum_{i=1}^{K_U} x_i^U, \quad j_c = \frac{1}{K_U} \sum_{i=1}^{K_U} j_i^U.
\]

(2) When \(T \to +\infty\), the UAV would have enough time that it can reach the top of each GU to enjoy the employ communication channel. As \(T\) is sufficiently large, we could neglect the UAVs’ time to fly among different users. Then \(T\) is divided between all users for the UAV to hover on. For fairness, the hovering time \(T_i\) distributed to user \(k_i^U\) is inversely proportional to its communication rate \(r_i^h\):

\[
r_i^h = \log_2 \left[ 1 + \frac{P_t^U b_0}{(P_t^U g_{x_i}^U + \mathcal{B}N_i)\|\mathbf{q}_i^U - \mathbf{q}_0^U\|^2} \right] \geq \frac{\mathcal{R}_{\min}^U}{B},
\]

\[
T_i = \frac{T - T_{\text{trav}}}{\sum_{i=1}^{K_U} 1 / r_i^h}. \quad (38)
\]

We obtain two trajectory initialisation schemes with low complexity based on the two extreme cases called ‘circle-init’ and ‘TSP-init’. In circle-init, the UAV would fly around the point \(\mathbf{q}_{0i}^U\), and the trajectory is designed as a circle centred on \(\mathbf{q}_{0i}^U\). The circle’s circumference is set as \(L_{\text{max}}^U T\) when \(T\) is small. As \(T\) increases, the circle’s radius would reach \(r_{\text{max}}^U\), which is the maximum value for all circles centred on \(\mathbf{q}_{0i}^U\) and confined in the map.

In TSP-init, the UAV tries to fly over each associated GUs and hover on them. As the actual period \(T\) is limited, we have to minimise the UAV travelling time \(T_{\text{trav}}\) to set aside more time for hovering. Considering that the UAV’s maximum horizontal speed \(V_{\text{max}}^U\) is constant, minimising \(T_{\text{trav}}\) is equivalent to minimising the total travelling distance among different GUs. The latter is known as the classic travelling salesman problem (TSP), where a list of cities and the distances between each pair of cities are given, and the shortest possible route that visits each city exactly once and returns to the origin city needs to be found. Although TSP is a non-deterministic polynomial complete problem, we could solve it efficiently by dynamic programming in our small-scale case (number of GUs connected to the UAV is not large). If \(T > T_{\text{trav}}\), we could take the TSP problem’s solution as the UAV’s flight trajectory. The hover time is divided like above:

\[
T_i = \frac{(T - T_{\text{trav}}) / r_i^h}{\sum_{i=1}^{K_U} 1 / r_i^h}. \quad (39)
\]

But if \(T \leq T_{\text{trav}}\), then there is no time left for hovering. To design a closed trajectory, we scale down the TSP problem’s solution to ensure that the UAV can exactly fly one round in the period \(T\). The central point adopted as the reference point of re-scaling is \(\mathbf{q}_{0i}^U\).

As we derive circle-init and TSP-init from the two extreme cases, they are not universal. While circle-init does not suit the case when \(T\) is large, TSP-init does not suit when \(T\) is small. Therefore, we combine these two schemes as a new scheme named ‘comb-init’. Comb-init is employed as the final initialisation scheme, where we invoke TSP-init when \(T > T_{\text{trav}}\), and invoke circle-init when \(T \leq T_{\text{trav}}\).

### 3.1.4 Overall algorithm

In the final, we obtain an iterative algorithm that successively solves the subproblems presented above by the BSUM method. For simplicity, we adopt the cyclic rule to select the variables to be optimised in each iteration. As problem (24) can be solved exactly, and problem (34) needs successively approximating, we first solve the user association problem for once and then conduct the UAV trajectory optimisation for \(K_{\text{trav}}\) times in a cycle of iteration. Note that the solution we acquire by LP (24) is continuous, but in actual applications, the value of \(a_{k}^{U}[n]\) and \(a_{k}^{U}[v]\) can only be 0 or 1. To obtain the integral solution for \(a_{k}^{U}[n]\) and \(a_{k}^{U}[v]\), we simply have the continuous solution rounded at last. The whole algorithm is summarised in Algorithm 1.
3.2 | BSUM-based penalty algorithm

In this subsection, an algorithm applying the penalty factor based on the BSUM algorithm is proposed, called the 'BSUM-based penalty algorithm'. Meanwhile, a penalty factor is introduced to restrict \(a^G_k[a], a^U_k[a]\) to near 0 or 1.

3.2.1 Penalty factor

Unlike the first algorithm, a penalty factor is added to the objective function after the relaxation of the binary constraints. By introducing the penalty factor, \(a^G_k[a]\) and \(a^U_k[a]\) would be restricted to near 0 or 1 while the algorithm searches for more massive throughput. Therefore, this algorithm is more similar to the actual problem in the structure, where the values of \(aG[n], aU[n]\) are binary. Besides this, the changes in \(a^G_k[a], a^U_k[a]\) between different iterations also become much more extensive.

In the relaxed BSUM algorithm, the searching areas of \(a^G_k[a], a^U_k[a]\) are mainly limited by the gradients of \(R_{\min}\). When the penalty factor is applied, the objective function gradients are added with an item that inspires the algorithm to move towards those points whose \(a^G_k[a], a^U_k[a]\) are all 0 or 1, and helps the algorithm out of local maximum points. This advantage is validated in the numerical simulations.

Function \(P_u(x) = \frac{1}{1 + F(1 - x)}\) is chosen to form our penalty factors, with which we get a lower bound of \(R_k^U[a]\):

\[
R_k^U[a] = \frac{1}{N} \sum_{i=1}^{N_i} \left[ \frac{BE_k^G[a]a^G_k[a]}{1 + F(1 - a^G_k[a])} + \frac{BE_k^U[a]a^U_k[a]}{1 + F(1 - a^U_k[a])} \right].
\]  

(40)

As our optimisation target is the minimum one between the \(K\) items of \(R_k^U[a]\), replacing \(R_k^U[a]\) with \(R_k^L[a]\) is an equivalent way to multiply the objective function with the penalty factors.

If coefficient \(F\) is sufficiently large, \(R_k^L[a]\) has a considerable value only when \(a^G_k[a]\) or \(a^U_k[a]\) are near 1. Otherwise, the corresponding item would be 'penalised' to nearly 0. It is worth noting that there are factors \(a^G_k[a], a^U_k[a]\) in \(R_k^L[a]\), so the penalty factor does not affect the situations when \(a^G_k[a], a^U_k[a] = 0\). To maximise \(R_k^L[a]\), the algorithm seeks to make more \(a^G_k[a], a^U_k[a]\) valued at 1 under constraints (11) \sim (13). Finally, we obtain the integral solutions of \(a^G_k[a]\) and \(a^U_k[a]\) with high efficiency.

3.2.2 Subproblem of user association

Assuming the UAV’s trajectory is obtained, \(E_k^G[a]\) and \(E_k^U[a]\) are considered to be constants. Expression (40) is not convex or concave, so we apply the SCA method in [27] to approximate it. With the linear approximation

\[
F'(x, x_0) = \frac{1 + F}{[1 + F(1 - x_0)]^2} (x - x_0) + \frac{x_0}{1 + F(1 - x_0)}
\]

a lower bound of (40) is obtained as

\[
R_k^P[a] = \frac{B}{N} \sum_{i=1}^{N_i} \left[ F'(a^G_k[a], a^G_r'[a])E_k^G[a] + F'(a^U_k[a], a^U_r'[a])E_k^U[a] \right],
\]

(42)

and the user association problem is approximated as P7

\[
P7: \begin{align*}
\max_{a^G_k[a], a^U_k[a], R_{\min}} & \quad R_{\min} \\
\text{s.t.} & \quad \frac{B}{N} \sum_{i=1}^{N_i} \left[ F'(a^G_k[a], a^G_r'[a])E_k^G[a] + F'(a^U_k[a], a^U_r'[a])E_k^U[a] \right] \geq R_{\min}, \quad \forall k, \\
& \quad a^G_k[a], a^U_k[a] \in [0, 1], \quad \forall k, n
\end{align*}
\]

(43a)

(43c)

(11) \sim (13), where \(a^G_r'[a], a^U_r'[a]\) are local points. For the first iteration, \(a^G_r[a], a^U_r[a]\) are obtained by solving the LP problem P3.

In the \(i\)th iteration, \(a^G_r[a], a^U_r[a]\) are succeeded from the last iteration results. The results are stored as \(a^{G,r_i+1}[a], a^{U,r_i+1}[a]\) for the use of the next iteration. After several iterations, \(a^G_r[a], a^U_r[a]\) would be all confined near 0 or 1.

3.2.3 Subproblem of UAV trajectory optimisation

In this subproblem, values of \(a^G_r[a], a^U_r[a]\) are all considered to be binary constants. Therefore, the subproblem of UAV trajectory optimisation with penalty factors is the same as the problem (25) and disposed of in the same method.

3.2.4 Overall algorithm

An iteration strategy similar to the BSUM algorithm is adopted here, which also alternatively solves user association and trajectory optimisation subproblems. A difference from the first algorithm is that the subproblem of user association is no longer a simple LP program, but added with the penalty factors and solved by the SCA approximation. Details are described in Algorithm 2.
3.3 Convergence analysis

This subsection analyses our algorithms’ convergence behaviours and discusses the relationship between the original problem and the transformed problems. The first conversion is relaxation from the original problem (P1) to (P2). In this conversion, the objective function remains the same, represented as $\Gamma(A, Q)$. Denoting the optimisation values of problems (P1) and (P2) as $R_{ori}, R_{rel}$, then there is $R_{ori} \leq R_{rel}$ as the relaxation operation enlarges A’s and Q’s feasible region. Then problem (P2) is partitioned into two subproblems (P3) and (P4) in both algorithms, whose optimisation values are lower bounds of $R_{rel}$ because the feasible area is narrowed down when A or Q is held.

In the relaxed BSUM algorithm, the user association subproblem (P3) is precisely processed by LP. If the value of $\Gamma(A, Q)$ is $\Gamma'(A', Q')$ before the rth iteration of user association optimisation, then it becomes $\Gamma'(A'^{r+1}, Q'^{r})$ after this iteration. As this LP problem is globally solved, we have

$$\Gamma(A', Q') \leq \Gamma'(A'^{r+1}, Q'^{r}).$$

(44)

The UAV trajectory subproblem (P4) is approximated by the problem (P5), where the objective function is replaced by $\hat{\Gamma}'(A, Q)$. $\hat{\Gamma}'(A, Q)$ is a global lower bound of $\Gamma(A, Q)$, and satisfies

$$\hat{\Gamma}'(A'^{r+1}, Q'^{r}) = \hat{\Gamma}'(A'^{r+1}, Q'^{r}).$$

(45)

Problem (P5) is also solved precisely, and the optimal objective value would therefore be no less than the function value at the local point. Also note that $\hat{\Gamma}'(A, Q)$ is a global lower bound of $\Gamma(A, Q)$, then there is

$$\hat{\Gamma}'(A'^{r+1}, Q'^{r}) \leq \hat{\Gamma}'(A'^{r+1}, Q'^{r+1}) \leq \Gamma'(A'^{r+1}, Q'^{r+1}).$$

(46)

Therefore, we have proved that $\Gamma(A', Q') \leq \Gamma'(A'^{r+1}, Q'^{r+1})$. Because $R_{rel}$ is an upper bound of all subproblems’ optimal objective value, we could know that $\Gamma'(A', Q')$ would converge to a particular value $R_{int}$ as $r$ increases. Influenced by the problem of local maximum points, there may be a massive difference between $R_{con}$ and $R_{rel}$. Finally, the converged continuous solution is rounded to an integral solution $A'^{int}, Q'^{int}$. Meanwhile, the corresponding system throughput would fall to $R_{int} = \Gamma(A'^{int}, Q'^{int}) \leq R_{con}$.

In the BSUM-based penalty algorithm, there are several penalty-based optimisations after solving the LP problem in user association. These optimisations force variables A to be near binary, while they might lower the throughput, like the performance loss caused by the rounding operation in the relaxed BSUM algorithm. Therefore, this algorithm may keep vibrating rather than converge to one point. Fortunately, the simulation results indicate that this algorithm also roughly converge and that the vibration amplitude is pretty small.

Our analysis of the algorithms’ convergence is validated in numerical simulations and visually described in Figure 2 with a concrete case. The optimal values of problems (P1) and (P2) are drawn schematically, as they cannot be obtained from our algorithms.

4 SIMULATION RESULTS

In this section, the system performance of our proposed schemes is evaluated through numerical simulations. A rectangular area of size $3 \text{ km} \times 3 \text{ km}$ is considered, in which the GBS is fixed 20 m above the centre point. For comparison, the GUs and
TABLE 1 Parameter values for numerical simulations

| Parameter                        | Value          |
|----------------------------------|----------------|
| GBS altitude                     | \( H_{G} = 20 \text{ m} \) |
| Uniform GUs’ formation density   | \( \lambda_u = 5/9 \text{ km}^{-2} \) |
| Non-uniform GUs’ formation density | \( \lambda_{nu} = 10/9 \text{ km}^{-2} \) |
| Shift factor of non-uniform GUs   | \( \rho = 0.6 \) |
| UAV flight period                 | \( T = 240 \text{ s} \) |
| Element slot length               | \( \delta_e = 0.8 \text{ s} \) |
| Maximum UAV height                | \( H_{\text{max}} = 1200 \text{ m} \) |
| Minimum UAV height                | \( H_{\text{min}} = 200 \text{ m} \) |
| UAV’s height for initialisation   | \( H_{0} = 300 \text{ m} \) |
| Maximum horizontal speed of UAV   | \( V_{\text{max}}^{h} = 30 \text{ m/s} \) |
| Maximum vertical speed of UAV     | \( V_{\text{max}}^{v} = 15 \text{ m/s} \) |
| Transmit power of GBS             | \( P_{G}^{0} = 31 \text{ dBm} \) |
| Transmit power of UAV             | \( P_{U}^{p} = 0.12 \text{ W} \) |
| Frequency spectrum bandwidth      | \( B = 1 \text{ MHz} \) |
| Reference GBS–GU channel power gain at \( d_{b} = 1 \text{ m} \) | \( g_{b} = -19.6 \text{ dB} \) |
| Path loss coefficient of GBS–GU channel | \( s = 3.9 \) |
| Standard deviation of decibel loss by shadow fading | \( \sigma = 8 \text{ dB} \) |
| Reference UAV–GU channel power gain at \( d_{u} = 1 \text{ m} \) | \( g_{u} = 1 \times 10^{-6} \) |
| Noise power spectrum density      | \( N_{0} = -170 \text{ dBm/Hz} \) |
| Threshold received power in initialisation scheme | \( P_{th} = -100 \text{ dBm} \) |
| The convergency threshold of Algorithm 1 | \( \varepsilon_{1} = 1 \times 10^{-5} \) |
| The convergency threshold of Algorithm 2 | \( \varepsilon_{2} = 2 \times 10^{-2} \) |
| Successive iteration times of (34) in Algorithm 2 | \( k_{\max} = 10 \) |
| Successive iteration times of (41) in Algorithm 2 | \( k_{\text{cha}} = 10 \) |

Abbreviations: GBS, ground base station; GU, ground user; UAV, unmanned aerial vehicle.

channel parameters are generated by one random realisation of our program. All parameters used to construct the system and execute the algorithms are assigned as Table 1, except for those we specially stated in the following text or figures. All the simulations are conducted in the MATLAB environment applying the CVX, a package for specifying and solving convex programs [31].

To validate of our two proposed algorithms’ effectiveness, they are compared with a fundamental scheme in which the user association scenarios of the GBS and UAV are optimised separately. In the fundamental scheme, GUs are divided into two groups, like in the initialisation scheme. The GUs with sufficient \( P_{G}^{G_{i,k}} \) are served by the GBS in turn, and the others are distributed to the UAV. Interference is considered in the fundamental scheme, while UAV’s trajectory and user association scenarios are jointly optimised.

4.1 Iteration progress

The convergence behaviours of our proposed algorithms are described in Figure 2. The iteration times of trajectory optimisation that consumes the main part of computation time and resource is taken as the indicator to describe the algorithms’ iteration progress. The blue curve depicts the continuous solution we obtain in the relaxed BSUM algorithm, while the red curve shows the rounded solution, that is, the final solution. Differences between these two curves reveal the rounding operation’s performance loss, which is about 2%. The black curve denotes the BSUM-based penalty algorithm. We can conclude that the relaxed BSUM algorithm converges quickly and stably and that the BSUM-based penalty algorithm converges slowly and keeps fluctuating. However, the latter performs much better when it comes to max–min communication throughput. These results verified our discussion in Subsections 3.2 and 3.3. In the application, the relaxed BSUM algorithm is more appropriate for cases demanding low latency, while the BSUM-based penalty algorithm satisfies higher throughput requirements.

4.2 Optimisation results

In this subsection, the output results of different algorithms are compared in Figures 3 and 4. Intuitonally, the user association schemes and UAV trajectory generated by the relaxed BSUM algorithm and the BSUM-based penalty algorithm look alike. Therefore, we just display the results of the latter to save space. The results obtained from the fundamental scheme are also shown in comparison.

The user association schemes are represented in Figure 3, where the GUs are sorted in ascending order by the power
In the fundamental scheme, the GUs are rigidly divided into two groups by $P_{r,k}$. The GBS chooses those GUs with higher $P_{r,k}$ as its users, while the UAV serves the others. In this way, the UAV could assist those GUs who cannot communicate with the GBS efficiently. In addition, this method could also mitigate the co-channel interference as the GBS’s interference on these GUs is quite weak.

In the BSUM-based penalty algorithm, the association schemes also adopt the division method, but more flexible. There are several GUs that receive service from both the GBS and the UAV. As a result of the joint optimisation, user switch behaviours of the GBS and UAV are related. For example, there is a short-term user switch for both of them at the time of nearly 40 s. This flexibility provides a considerable benefit for the improvement of system throughput, which would be shown later.

Figure 4 depicts the UAV’s optimised 3D trajectory along with its horizontal projection. All GUs are represented as coloured dots, which reveals the quality of their communication channels to the GBS. Those GUs with colour closer to green have relatively better channels to the GBS, while colour closer to red means the opposite. In a whole period, the UAV flies across the map to provide service for GUs in different positions. Roughly, the UAV tries to fly near those GUs with lower $P_{r,k}$ while keeping a distance from the GU communicating with the GBS. When the UAV is close to its associated, it descends to utilise the best LOS channel. At other times, the UAV climbs up to avoid causing interference to the GBS’s user.

The UAV’s horizontal trajectories acquired from the BSUM-based penalty algorithm and the fundamental scheme are also compared. In Figure 4(a), the UAV approaches the top of each associated GU in turn. However, in Figure 4(b), the UAV has to fly near the map border sometimes. The reason is that the GBS’s user association schemes are not relevant to the UAV’s trajectory or user arrangements in the fundamental scheme, and the UAV has to fly away if a nearby GU is communicating with the GBS. While in our BSUM-based penalty algorithm, the GBS’s user association schemes are jointly optimised with other variables. Therefore, the GBS could select a GU far from the UAV for communicating, rather than keeping the UAV away.

4.3 | Performance analysis

To validate our proposed algorithms’ effectiveness, we first consider the fairness between different GUs via Figure 5. The average throughput of each GU is displayed as the green bars, and the minimum average throughput among all GUs is represented as the black dashed line. This figure is drawn using the BSUM-based penalty algorithm’s outputs when the UAV’s flight period $T = 240$ s, where the relative difference between the maximum and minimum average throughput values is 2.5%. When $T$ is more prolonged, this indicator could descend to 1% due to the increased flexibility. The relaxed BSUM algorithm also shows a similar performance. These results demonstrate that our method of applying the minimum average throughput $R_{\text{min}}$ as the optimisation objective is quite beneficial to user fairness.

Then the performance of our initialisation schemes circle-init, TSP-init, and comb-init are compared in Figure 6. As
we predicted, circle-init outputs greater throughput when $T$ is small, while TSP-init performs better when $T$ is large. The combined algorithm comb-init accurately select and utilise the better algorithm in all tested cases.

Besides, we also analysed the impacts of the UAV's height on the system performance by fixing the UAV's height in the algorithms. The results are displayed in Figure 7, where the 3D algorithms' performance is also shown as references. The height of the UAV influences the network in two aspects. On the one hand, a lower altitude could provide the UAV with better channels to communicate with GUs, thus is beneficial to the network. On the other hand, the users linked to the GBS would suffer more serious interference if the UAV flies down, which will worsen the system performance. These influences are apparent in the fundamental scheme, whose throughput first climbs up, but then falls down as the UAV's height increases. In our BSUM-based algorithm and the relaxed BSUM algorithm, the interference problem's influence is pretty reduced via the joint optimisation of trajectory and the UAV and GBS's user association schemes. Therefore, the impacts on the UAV's communication channel would play a dominant role and keep improving the system performance as the UAV's height decreases to $H_{\text{min}}$. Anyhow, the 3D schemes show better performance than 2D schemes in all algorithms.

Figure 8 describes the throughput results obtained at different $T$. The performance of our two proposed algorithms and the fundamental scheme is contrasted with their 2D versions. Note that the UAV's height is also optimised to maximise the system throughput in the 2D schemes. In the BSUM-based penalty algorithm and the relaxed BSUM algorithm, the 3D versions only have small improvements upon the 2D versions. This is because the interference impact has been weakened a lot, as we stated above. Nevertheless, in the fundamental scheme, the 3D algorithm introduces a new optimisation dimension to relieve the co-channel interference, bringing a large gain. Moreover, this gain is enhanced as the increase of $T$, which allows the UAV to fly higher and more flexibly to avoid interference.

In the three 3D algorithms, the BSUM-based penalty algorithm behaves best, whose performance exceeds the fundamental one by around 20-30% in different cases. The throughput of the relaxed BSUM algorithm is similar to the fundamental scheme when $T$ is small. As $T$ increases, it improves rapidly and approaches the performance of the BSUM-based penalty algorithm.

Furthermore, the obtained max–min throughput grows with the increases of $T$ in all algorithms, as the longer $T$ provides more possible combinations of the user association schemes and a broader flight range. However, a longer $T$ also means that a GU needs to wait more time to acquire service, and this is commonly referred to as the rate-delay balancing problem. When $T$ is sufficiently large, the UAV could reach the map boundary area, and the user association scheme is near optimal, leading to the saturation of the throughput. Besides, the saturation values of the throughput in the 3D and 2D versions of the relaxed BSUM algorithms and the BSUM-based algorithms are close, while the 3D and 2D versions of the fundamental scheme have similar saturation values.
5 | CONCLUSION

This paper proposes a design to reduce the interference in a UAV-assisted hybrid cellular network by choosing the suitable UAV trajectory and user association schemes. Specifically, the UAV and GBS’s user association, along with UAV’s 3D trajectory, are jointly optimised to maximise the minimum average throughput between all GUs with the existence of several constraints. By applying the framework of BSUM and introducing the penalty factor, two effective algorithms are, respectively, raised for application of low latency and QoS scenes. A method to simplify the calculation of logarithmic expressions in the optimisation problem is also put forward. Our proposed algorithms are numerically verified to be effective and outperform the fundamental scheme to a great extent.

ACKNOWLEDGEMENT

This work was supported by the president’s fund for undergraduate students at Peking University.

REFERENCES

1. Li, B., et al.: UAV communications for 5G and beyond: recent advances and future trends. IEEE Internet Things J. 6(2), 2241–2263 (2019)
2. Zhang, H., et al.: Unmanned Aerial Vehicle Applications over Cellular Networks for 5G and Beyond. Springer, Berlin (2020)
3. Zhang, J., et al.: Altitude and number optimisation for UAV-enabled wireless communications. IET Commun. 14(8), 1228–1233 (2020)
4. Chen, Q.: Joint trajectory and resource optimization for UAV-enabled relaying systems. IEEE Access 8, 24108–24119 (2020)
5. Zhang, S., et al.: Joint trajectory and power optimization for UAV relay networks. IEEE Commun. Lett. 22(1), 161–164 (2018)
6. Zeng, Y., et al.: Trajectory design for completion time minimization in UAV-enabled multicasting. IEEE Trans. Wireless Commun. 17(4), 2233–2246 (2018)
7. Hua, M., et al.: Energy-efficient optimisation for UAV-aided wireless sensor networks. IET Commun. 13(8), 972–980 (2019)
8. Wang, H., et al.: Power control in uav-supported ultra dense networks: communications, caching, and energy transfer. IEEE Commun. Mag. 56(6), 28–34 (2018)
9. Zhang, S., et al.: Beyond D2D: full dimension UAV-to-everything communications in 6G. IEEE Trans. Veh. Technol. 69(5), 6592–6602 (2020)
10. Zhang, S. et al.: Cellular UAV-to-X communications: design and optimization for multi-UAV networks. IEEE Trans. Wireless Commun. 18(2), 1346–1359 (2019)
11. Xu, J., et al.: UAV-enabled wireless power transfer: trajectory design and energy optimization. IEEE Trans. Wireless Commun. 17(8), 5092–5106 (2018)
12. Sharma, V., et al.: UAV-assisted heterogeneous networks for capacity enhancement. IEEE Commun. Lett. 20(6), 1207–1210 (2016)
13. Zhang, X., et al.: Joint trajectory and power optimization for UW sensing over cellular networks. IEEE Commun. Lett. 22(11), 2382–2385 (2018)
14. Mei, W., et al.: Cellular-connected UAV uplink association, power control and interference coordination. IEEE Trans. Wireless Commun. 18(11), 5380–5393 (2019)
15. Chu, E., et al.: Interference modeling and analysis in 3-dimensional directional UAV networks based on stochastic geometry. IET Express 5(4), 235–239 (2019)
16. Liu, L., et al.: Multi-beam UAV communication in cellular uplink: cooperative interference cancellation and sum-rate maximization. IEEE Trans. Wireless Commun. 18(10), 4679–4691 (2019)
17. Challita, U., et al.: Interference management for cellular-connected UAVs: a deep reinforcement learning approach. IEEE Trans. Wireless Commun. 18(4), 2125–2140 (2019)
18. Hu, J., et al.: Reinforcement learning for a cellular internet of UAVs: protocol design, trajectory control, and resource management. IEEE Wireless Commun. 27(1), 116–123 (2020)
19. Wu, F., et al.: Cellular UAV-to-device communications: trajectory design and mode selection by multi-agent deep reinforcement learning. IEEE Trans. Commun. 68(7), 4175–4189 (2020)
20. Zhang, H., et al.: Cooperation techniques for a cellular internet of unmanned aerial vehicles. IEEE Wireless Commun. 26(5), 167–173 (2019)
21. Rahmati, A., et al.: Dynamic mobility-aware interference avoidance for aerial base stations in cognitive radio networks. In: IEEE INFOCOM 2019 – IEEE Conference on Computer Communications, pp. 595–603. IEEE, Paris (2019)
22. Wu, Y., et al.: Throughput maximization of UAV-enabled wireless network in the presence of jammers: joint trajectory and communication design. IEEE Trans. Commun. E102B (10), 1983–1990 (2019)
23. Shen, C., et al.: Multi-UAV interference coordination via joint trajectory and power control. IEEE Trans. Signal Process. 68, 843–858 (2020)
24. Lyu, J., et al.: UAV-aided offloading for cellular hotspot. IEEE Trans. Wireless Commun. 17(6), 3988–4001 (2018)
25. Hong, M., et al.: A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing. IEEE Signal Process Mag. 33(1), 57–77 (2016)
26. Li, C., et al.: Mixed spatial traffic modeling of heterogeneous cellular networks. 2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB). pp. 1–5. IEEE, Montreal (2015)
27. Razaviyayn, M.: Successive convex approximation: analysis and applications. PhD thesis, University of Minnesota (2014)
28. Yuille, A., Rangarajan, A.: The concave–convex procedure. Neural Comput. 15(4), 915–936 (2003)
29. Zhang, J., et al.: UAV-enabled radio access network: multi-mode communication and trajectory design. IEEE Trans. Signal Process. 66(20), 5269–5284 (2018)
30. Xu, Z., et al.: Join trajectory optimization and communication design for UAV-enabled OFDM networks. Ad Hot Networks 98, 102031 (2020)
31. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx (2018). Accessed December

APPENDIX

Remark 1
For $x \in [0, +\infty)$, the first and second derivatives of $f(x) = \ln(1 + x)$ are

$$f^{(1)}(x) = \frac{1}{1 + x} > 0,$$

$$f^{(2)}(x) = -\frac{1}{(1 + x)^2} < 0.$$
Therefore, \( f(x) \) is a concave and monotonically increasing function. A concave fractional function could be applied to approximate \( f(x) \) near local point \( x_0 \):

\[
g(x, x_0) = C(x_0) - \frac{1}{a(x_0)x + b(x_0)}.
\]

The following constraints are imposed on \( g(x, x_0) \) to ensure that \( g(x, x_0) \) is a global lower bound of \( f(x) \) in \([0, +\infty)\) and that \( g(x, x_0) \) is a tight approximation near the local point \( x_0 \):

\[
\begin{align*}
g(x_0, x_0) &= f(x_0) \quad (A.1a) \\
g^{(1)}(x, x_0) &\geq f^{(1)}(x), \quad x < x_0 \quad (A.1b) \\
g^{(1)}(x, x_0) &= f^{(1)}(x), \quad x = x_0 \\
g^{(1)}(x, x_0) &\leq f^{(1)}(x), \quad x > x_0.
\end{align*}
\]

By solving (A.1), the constraints on coefficients \( a(x_0), b(x_0), C(x_0) \) are obtained as

\[
\frac{x_0 + 2 - 2\sqrt{x_0} + 1}{x_0^2} \leq a(x_0) \leq \frac{x_0 + 2 + 2\sqrt{x_0} + 1}{x_0^2} \quad (A.2a)
\]

\[
b(x_0) = \sqrt{a(x_0)(x_0 + 1)} - a(x_0)x_0 \quad (A.2b)
\]

\[
C(x_0) = \ln (1 + x_0) + \frac{1}{a(x_0)x_0 + b(x_0)}. \quad (A.2c)
\]

Then the first derivative of \( g(x, x_0) \) is

\[
g^{(1)}(x, x_0) = \frac{a(x_0)}{[a(x_0)x + b(x_0)]^2} = \frac{1}{\left[\sqrt{a(x_0)(x - x_0)} + \sqrt{x_0 + 1}\right]^2}.
\]

Through several computation, we could reach a conclusion that the estimation error of \( g(x, x_0) \) on \( f(x_0) \) becomes smaller when \( a(x_0) \) degrades. Therefore, the best value of \( a(x_0) \) is

\[
a(x_0) = \frac{x_0 + 2 - 2\sqrt{x_0} + 1}{x_0^2},
\]

and the values of \( b(x_0), C(x_0) \) are given by Equations (A.2b) and (A.2c).