User Association and Path Planning for UAV-Aided Mobile Edge Computing with Energy Restriction

Yuwen Qian, Feifei Wang, Jun Li, Long Shi, Kui Cai, and Feng Shu

Abstract—Mobile edge computing (MEC) provides computational services at the edge of networks by offloading tasks from user equipments (UEs). This letter employs an unmanned aerial vehicle (UAV) as the edge computing server to execute offloaded tasks from the ground UEs. We jointly optimize user association, UAV trajectory, and uploading power of each UE to maximize sum bits offloaded from all UEs to the UAV, subject to energy constraint of the UAV and quality of service (QoS) of each UE. To address the non-convex optimization problem, we first decompose it into three subproblems that are solved with integer programming and successive convex optimization methods respectively. Then, we tackle the overall problem by the multi-variable iterative optimization algorithm. Simulations show that the proposed algorithm can achieve a better performance than other baseline schemes.

Index Terms—Mobile edge computing, UAV, user association, uploading power, trajectory optimization.

I. INTRODUCTION

Mobile edge computing (MEC) has been emerging as a promising technology that enables computationally complex applications at resource-limited user equipments (UEs) [1]. By offloading resource-consuming tasks and caching popular resources at the edge servers, MEC can greatly alleviate computational burden on the UEs and reduce data processing delay [2]. Furthermore, in rural and remote areas, it is not viable to deploy a large number of static servers for offloading tasks, due to complicated terrains and high costs. In these scenarios, mobile servers are more capable of combating uncertain environments than the static ones.

Among existing MEC strategies, the unmanned aerial vehicle (UAV) mounted MEC has attracted much attention because of its excellent maneuverability and low cost. In [3], the trajectory of a single UAV base station and user scheduling were jointly optimized to maximize the minimum average uploading rate of UEs, but it did not consider the battery capacities of UAVs. In practice, energy consumption of an UAV is a major issue when dealing with flight trajectory and
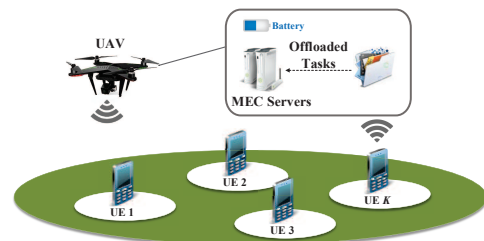
Fig. 1. The UAV-aided MEC system.

offloaded tasks. In light of this, a recent work [4] proposed an UAV-based MEC system that enables computing capabilities of the UAV to offer UEs with computation offloading services. This work aims at minimizing the overall energy consumption of the ground UEs under the UAV energy constraint by jointly optimizing task allocation and UAV trajectory.

In this letter, we study a novel UAV-aided MEC strategy to offer the ground UEs with efficient computational services. Our goal is to maximize sum bits offloaded from all UEs to the UAV subject to UAV energy constraint and quality of service (QoS) requirement of each UE. Toward this end, we jointly optimize user association, UAV trajectory, and uploading power of each UE. To solve this optimization problem, we decompose it into three separate subproblems, solve each subproblem by integer programming and successive convex optimization methods, and optimize the overall problem by applying multi-variable fixed iterative algorithm. Simulation results not only show that the UAV with optimized trajectory can improve the sum bits, but also reveal the impact of design parameters on the system performance, thereby providing guideline for the UAV deployment in MEC.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

As shown in Fig. 1, we consider that a single UAV provides computational services for K ground UEs, where the UAV flies over the UEs and receives the tasks. Meanwhile, the UAV with MEC servers executes these offloaded tasks from each UE and communicates with UEs to deliver the computational results. Also, UEs are distributed randomly each with a location $\mathbf{z}_k = (x_k, y_k), k \in \{1, 2, \cdots K\}$ over a 2-D coordinate plane.

Let T denote a task-offload period. During this period, the UAV departs from the origin, flies over the UEs for offloading tasks, and finally stops at a preset point $\mathbf{q}_T \equiv (x_T, y_T)$. We divide T into N equal time slots with the duration of each time slot being $\delta_t = \frac{T}{N}$. Considering that each time slot is sufficiently small, the UAV location in the n-th time slot is

$\mathbf{q}[n] = (x[n], y[n])$. Let $\{\mathbf{q}\} = \{\mathbf{q}[n], \forall n\}$ denote a UAV trajectory over N time slots with $\mathbf{q}[N] = \mathbf{q}_T$. In this letter, we only consider UAV flight altitude is fixed as a constant H. In addition, the UAV velocity $\nu[n]$ in the n-th time slot is constrained by the maximum velocity $\nu_{\max}$ as follows:

$$\nu[n] = \frac{\|\mathbf{q}[n+1] - \mathbf{q}[n]\|}{\delta_t} \leq \nu_{\max}, \forall n. \quad (1)$$

Let $b_k[n]$ be a binary variable of user association that indicates whether UE k is served by the UAV in the n-th time slot. If $b_k[n] = 1$, the UAV receives workload from UE k in the n-th time slot, otherwise $b_k[n] = 0$. At each time slot, the UAV serves at most one ground UE in the time division multiple access manner. These yield the following constraints:

$$\sum_{k=1}^{K} b_k[n] = 1, b_k[n] \in \{0, 1\}, \forall n, \forall k. \quad (2)$$

In this letter, we focus on the data uploading from the UEs to the UAV only and ignore latency caused by data downloading from the UAV to the UEs, as the size of computational results from the UAV is much less than that of offloaded tasks from the UEs. Consider that the channel between UAV and the UEs is line-of-sight (LoS) [3]. As such, the channel gain between UE k and the UAV in the n-th time slot is given by

$$h_k[n] = \frac{\rho_0}{H^2 + \|\mathbf{q}[n] - \mathbf{z}_k\|^2}, \forall n, \quad (3)$$

where ρ_0 denotes the channel gain at the reference distance of 1m and the 3-D distance from UE k to the UAV is $\sqrt{H^2 + \|\mathbf{q}[n] - \mathbf{z}_k\|^2}$. From (3), the uploading rate (bits/s/Hz) from UE k to the UAV is given by

$$R_k[n] = \log_2\left(1 + \frac{p_k[n] h_k[n]}{\sigma^2}\right), \forall n, \forall k, \quad (4)$$

where σ^2 is the power of additive white Gaussian noise (AWGN) at the UAV and $p_k[n]$ is the uploading power of UE k in the n-th time slot. Furthermore, we have

$$\sum_{n=1}^{N} p_k[n]\delta_t \leq E_U, \forall k, \quad (5)$$

$$p_k[n] \geq P_{\min}, \forall n, \forall k, \quad (6)$$

where (5) indicates that the total uploading energy of each UE over N time slots is upper bounded by the constant E_U and (6) shows that the minimum uploading power $P_{\min}$ of each UE to support the transmission of basic information to the UAV. From (4), the bits offloaded from each UE to the UAV over N time slots are given by

$$S_k = \left(\sum_{n=1}^{N} b_k[n] R_k[n]\right)\delta_t B, \forall k, \quad (7)$$

where B is the channel bandwidth. To meet the quality of service (QoS) of each UE, S_k yields

$$S_k \geq D_k, \forall k, \quad (8)$$

where D_k denotes the minimum offloaded bits of UE k to guarantee its QoS. The energy consumption of the UAV is mainly caused by flight and computation. On one hand, with reference to [4], the energy consumed by the UAV flight in the n-th time slot is $e_F[n] = \kappa\nu^2[n], \forall n$, where $\kappa = 0.5M\delta_t$ with

M being the UAV weight. On the other hand, the computation energy for UE k is given by

$$e_k = \gamma_C C_k S_k (f_C)^2, \forall k, \quad (9)$$

where C_k denotes the number of CPU cycles for computing every bit of UE k, f_C is the CPU frequency of the MEC servers, and γ_C is the effective switched capacitance [4].

Consider that the energy limit of UAV battery is E_0. We restrict the total energy consumption comprising $E_F = \sum_{n=1}^{N} e_F[n]$ and $E_C = \sum_{k=1}^{K} e_k$ to be $E_F + E_C \leq E_0$.

B. Problem Formulation

Define $\hat{S} = \sum_{k=1}^{K} S_k$ as the sum bits from all UEs to the UAV. On one hand, achieving higher uploading rate motivates the UAV to fly closer to the UE, which in turn consumes more flight energy. On the other hand, the more tasks the UEs offload, the more computation energy the UAV consumes. Driven by these observations, our objective is to maximize the sum bits subject to battery energy and QoS constraints. Towards this end, we jointly optimize the user association $\{b\}$, UAV trajectory $\{\mathbf{q}\}$, and UE uploading power $\{p\}$ as

$$\mathcal{P}: \max_{\{b\},\{\mathbf{q}\},\{p\}} \hat{S} \quad (10a)$$

$$\text{s.t.} \quad S_k \geq D_k, \forall k, \quad (10b)$$

$$E_C + E_F \leq E_0, \quad (10c)$$

$$\mathbf{q}[N] = \mathbf{q}_T, \quad (10d)$$

$$(1), (2), (5), \text{and } (6), \quad (10e)$$

where $\{b\} = \{b_k[n], \forall n, \forall k\}$ and $\{p\} = \{p_k[n], \forall n, \forall k\}$.

III. PROBLEM OPTIMIZATION

This section decouples $\mathcal{P}$ into three subproblems and solves the problem by the multi-variable fixed iterative algorithm.

A. User Association Optimization

Given the optimized UAV trajectory and uploading power of each UE (see line 3 of Algorithm 1 in Section III-D), this subproblem is to maximize the sum bits by optimizing the user association. As such, we can rewrite $\mathcal{P}$ as

$$\mathcal{P}_1: \max_{\{b\}} \hat{S} \quad (11a)$$

$$\text{s.t.} \quad (10b), (10c), \text{and } (2). \quad (11b)$$

Note that $\mathcal{P}_1$ is an integer programming problem. In this letter, we can solve $\mathcal{P}_1$ by branch and bound algorithm [5].

B. Trajectory Optimization

This part focuses on the trajectory design based on the optimized user association and uploading power of each UE (see line 4 of Algorithm 1). First, we can rewrite $\mathcal{P}$ as

$$\mathcal{P}_2: \max_{\{\mathbf{q}\}} \hat{S} \quad (12a)$$

$$\text{s.t.} \quad (10b), (10c), (10d), \text{and } (1). \quad (12b)$$

Note that $\mathcal{P}_2$ is an intractable problem due to the non-convex expression $R_k[n]$ with respect to $\mathbf{q}[n]$. However, it is convex

with respect to $\|\mathbf{q}[n] - \mathbf{z}_k\|^2$. Let $\{\mathbf{q}^r\} = \{\mathbf{q}^r[n], \forall n\}$ be the optimized trajectory in the r-th iteration (see Algorithm 1). Then we can replace $R_k[n]$ with its first-order Taylor expansion at $\mathbf{q}^r[n]$:

$$
\begin{aligned}
R_k[n] &= \log_2\left(1 + \frac{\rho_0 p_k[n]}{\sigma^2(H^2 + \|\mathbf{q}[n] - \mathbf{z}_k\|^2)}\right) \\
&\geq A_k^r[n]\left(\|\mathbf{q}[n] - \mathbf{z}_k\|^2 - \|\mathbf{q}^r[n] - \mathbf{z}_k\|^2\right) + W_k^r[n] \\
&= R_k^{\text{low},r}[n],
\end{aligned} \tag{13}
$$

where $W_k^r[n]$ and $A_k^r[n]$ are the coefficients of Taylor expansion of $R_k[n]$ at $\mathbf{q}^r[n]$ with respect to $\|\mathbf{q}[n] - \mathbf{z}_k\|^2$ and the inequality holds since any convex function is globally lower-bounded by its first-order Taylor expansion at any point [3]. Plugging (13) into (7), we further have

$$
S_k^{\text{low},r} = \sum_{n=1}^{N} \left(b_k[n] R_k^{\text{low},r}[n]\right) \delta_t B \tag{14}
$$

as the lower-bound approximation of S_k in (10b). Accordingly, $\hat{S}$ in (12a) can be replaced with $\hat{S}^{\text{low},r} = \sum_{k=1}^{K} S_k^{\text{low},r}$.

For the non-convex constraint of (10c), we derive a convex upper bound of non-convex function $R_k[n]$. For any $\mathbf{q}^r[n]$, the following convex approximation function is a global upper bound of $R_k[n]$ [6], given by

$$
\begin{aligned}
R_k^{\text{up},r}[n] &= W_k^r[n] + (\nabla_{\mathbf{q}[n]} R_k[n])_{\mathbf{q}^r[n]}^{\mathcal{T}}(\mathbf{q}[n] - \mathbf{q}^r[n]) \\
&\quad + \frac{L_{\nabla R_k[n]}}{2}\|\mathbf{q}[n] - \mathbf{q}^r[n]\|^2,
\end{aligned} \tag{15}
$$

where $(\nabla_{\mathbf{q}[n]} R_k[n])_{\mathbf{q}^r[n]}$ denotes the gradient of the non-convex function $R_k[n]$ at $\mathbf{q}^r[n]$ and $\mathcal{T}$ denotes the transpose. Notably, $(\nabla_{\mathbf{q}[n]} R_k[n])_{\mathbf{q}^r[n]}$ is Lipschitz continuous with constant $L_{\nabla R_k[n]} = \max\{\|\nabla^2 R_k[n]\|_2\}$ [7], where ∇^2 is Hessian matrix of $R_k[n]$ and $\|\cdot\|_2$ is the spectral norm of a matrix. Plugging (15) into (7) and (9), we can rewrite $\mathcal{P}_2$ as

$$
\mathcal{P}_2' : \max_{\{\mathbf{q}\}} \quad \hat{S}^{\text{low},r} \tag{16a}
$$
$$
\text{s.t.} \quad S_k^{\text{low},r} \geq D_k, \tag{16b}
$$
$$
E_{\text{C}}^{\text{up},r} + E_{\text{F}} \leq E_0, \tag{16c}
$$
$$
(10\text{d}), \text{ and } (1), \tag{16d}
$$

where $E_{\text{C}}^{\text{up},r}$ is the upper bound of E_{C} with respect to $R_k^{\text{up},r}[n]$. Therefore, $\mathcal{P}_2'$ becomes a convex problem, since (16a) and the left-hand-side (RHS) of (16b) are concave and the RHS of (16c) is convex. Finally, we solve $\mathcal{P}_2'$ by standard convex optimization techniques [8].

C. Uploading Power Optimization

Given the optimized user association and UAV trajectory (see line 5 of Algorithm 1), we can rewrite $\mathcal{P}$ as

$$
\mathcal{P}_3 : \max_{\{p\}} \quad \hat{S} \tag{17a}
$$
$$
\text{s.t.} \quad (10\text{b}), (10\text{c}), (5), \text{and } (6). \tag{17b}
$$

Let $p_k^r[n]$ be the optimized uploading power in the r-th iteration. Similar to (13) in $\mathcal{P}_2$, we can replace $R_k[n]$ with its first-order Taylor expansion at $p_k^r[n]$, because any concave function is globally upper-bounded by its first-order Taylor expansion at any point [3]. Then, E_{C} can be replaced by its

upper bound $E_{\text{C,power}}^{\text{up},r}$, which is a convex function with respect to $p_k^r[n]$. As such, we can rewrite $\mathcal{P}_3$ as

$$
\mathcal{P}_3' : \max_{\{p\}} \quad \hat{S} \tag{18a}
$$
$$
\text{s.t.} \quad E_{\text{C,power}}^{\text{up},r} + E_{\text{F}} \leq E_0, \tag{18b}
$$
$$
(10\text{b}), (5), \text{and } (6), \tag{18c}
$$

Note that $\mathcal{P}_3'$ is a convex problem and we solve it by the standard convex optimization method [8].

D. Overall Optimization

In this part, we solve the overall problem $\mathcal{P}$ by the multi-variable fixed iterative algorithm as shown in Algorithm 1.

Algorithm 1 Multi-variable fixed iterative algorithm

1: Set $r = 0$ and the tolerance error ϵ. Initialize UAV trajectory $\{\mathbf{q}^0\}$ and UE uploading power $\{p^0\}$.
2: **repeat**
3: Given $\{\mathbf{q}^r\}$ and $\{p^r\}$, find $\{b^{r+1}\}$ by solving $\mathcal{P}_1$;
4: Given $\{b^{r+1}\}$ and $\{p^r\}$, find $\{\mathbf{q}^{r+1}\}$ by solving $\mathcal{P}_2'$;
5: Given $\{b^{r+1}\}$ and $\{\mathbf{q}^{r+1}\}$, find $\{p^{r+1}\}$ by solving $\mathcal{P}_3'$;
6: Compute $\hat{S}^{r+1}$ using $\{b^{r+1}\}$, $\{\mathbf{q}^{r+1}\}$ and $\{p^{r+1}\}$.
7: $r \leftarrow r + 1$.
8: **until** $|\hat{S}^{r+1} - \hat{S}^r| \leq \epsilon$.

IV. NUMERICAL RESULTS

In this section, we evaluate performance of the proposed algorithm. We set a period of $T = 120$s with $N = 50$ time slots. The number of the UEs is $K = 8$. The UAV altitude is $H = 50$m and its weight is $M = 10$kg. The noise power is $\sigma^2 = -140$dB. The channel bandwidth is $B = 10$MHz. The channel gain at the reference distance is $\rho_0 = -50$dB. The number of CPU cycles C_k for each UE is shown in the data box of Fig. 2. We set $E_{\text{U}} = 36$J in (5) and initial $p_k[n] = 0.3$W in (6). The effective switched capacitance is $\gamma_{\text{C}} = 10^{-27}$ [9] and the initial trajectory is set as a ring.

Fig. 2 depicts the optimized trajectories under different battery energy, where the stars illustrate the locations of the UEs. The data box shows the minimum offloaded bits D_k and the bits S_k from UE k when battery energy $E_0 = 360$kJ. First, larger battery energy contributes to larger UAV coverage area. When E_0 is small, e.g., $E_0 = 120$kJ, the UAV flies towards the UEs closer to the initial trajectory such as UEs 5 and 8, in order to save flight energy. Conversely, when E_0 is sufficiently large, e.g., $E_0 = 360$kJ, the UAV can reach the location of each UE to obtain higher uploading rate. Second, from the density of sampling points, we find that the UAV reduces the speed to collect more tasks as it flies closer to the UE. For example, the velocity $\nu' < \nu$, as it is observed that the UAV flies different distances over the same period of time. Third, from D_k and S_k in the data box, the UEs closer to the initial trajectory offload more bits than those farther from the trajectory. For example, $S_7 > S_6$ as UE 7 is closer to the initial trajectory than UE 6. Fourth, we observe that larger C_k results in less offloaded bits due to higher computational complexity. For example, $S_1 < S_5$ as $C_1 > C_5$.
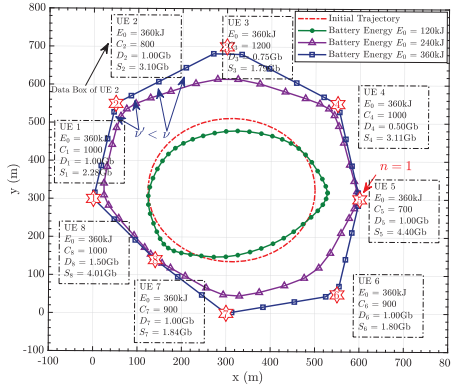
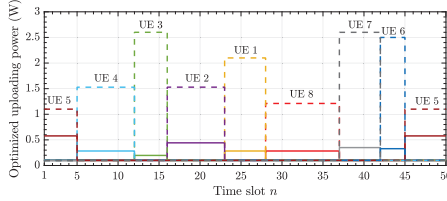Fig. 2. Optimized trajectory of UAV-aided MEC with $f_C = 2$GHz and $\nu_{\max} = 30$m/s.

Fig. 3. Optimized uploading power $p_k[n]$ versus time slots, where the dashed and solid lines correspond to $E_0 = 360$kJ and $E_0 = 120$kJ respectively.

Fig. 4. The UAV velocity $\nu[n]$ versus time slots.

Fig. 5. The sum bits $\hat{S}$ versus battery energy E_0 with $D_k = 0.1$Gb.

Fig. 3 shows the optimized uploading power for all UEs in each time slot based on the optimized trajectory in Fig. 2, where the dashed and solid lines correspond to $E_0 = 360$kJ and $E_0 = 120$kJ respectively. Lines with different colors represent different UEs. First, the UE keeps the uploading power invariant when it is served by the UAV, while the UE maintains the minimum communication power $P_{\min} = 0.1$W if it is not served. That is because the channel gain between the UAV and its served UE is almost unchanged during its service period. Second, the uploading power of each UE increases as E_0 rises when it is served by the UAV, since each UE can upload more bits to the UAV with larger computation energy. Third, UE 3 has the highest uploading power when it is served among all UEs. This is due to the fact that UE 3 requires the largest computation energy for each bit (see C_3 in Fig. 2) and its location is relatively farther from the initial trajectory.

Fig. 4 depicts the UAV velocity in each time slot versus maximum velocity under sufficient battery energy $E_0 = 360$kJ. The other parameters are the same with Fig. 2. First, when $\nu_{\max} = 10$m/s, the UAV needs to fly at this $\nu_{\max}$ over all time slots to serve each UE. Second, when $\nu_{\max} = 30$m/s, the UAV first flies towards the UE quickly at this $\nu_{\max}$, and then it reduces speed to collect more tasks as it reaches the location of each UE as Fig. 2 shows. Third, from each data box, we observe that increasing maximum velocity not only increases sum bits $\hat{S}$ but also consumes more flight energy.

Fig. 5 examines the impact of battery energy E_0 on the sum bits $\hat{S}$, where Scheme I optimizes $\mathcal{P}_1$ and $\mathcal{P}_2$ under fixed UE uploading power [3], and Scheme II optimizes $\mathcal{P}_1$ under fixed UAV trajectory and UE uploading power. First, the proposed algorithm improves $\hat{S}$ by 16.8% and 37.3% compared with Scheme I and Scheme II respectively. Second, $\hat{S}$ goes up as E_0 increases and $\hat{S}$ reaches the peak due to the limited uploading power. For example, for the proposed algorithm at $f_C = 1$GHz, $\hat{S}$ rises as E_0 goes up and retains the peak at
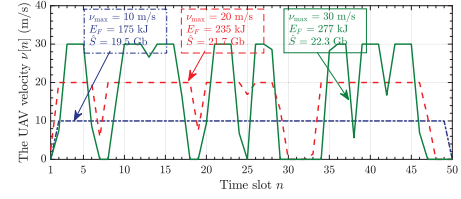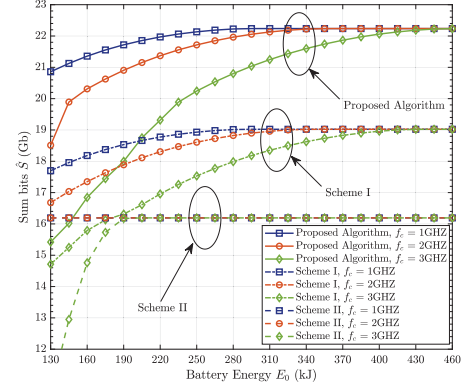
22.2Gb when $E_0 \geq 280$kJ. Third, for any fixed battery energy, $\hat{S}$ drops as f_C increases. This is caused by the fact that the increase of f_C consumes more computing energy.

V. CONCLUSION

We have optimized user association, the trajectory of UAV and UE uploading power to maximize the sum bits of offloaded tasks in the UAV-aided MEC system subject to the UAV battery energy and QoS constraints, using integer programming and successive convex optimization methods. Moving forward, it is of interest to study dynamic MEC systems where wireless channels, user locations, and UE tasks evolve in real time.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, May 2017.

[2] J. Li, H. Chen, Y. Chen, Z. Lin, B. Vucetic, and L. Hanzo, "Pricing and resource allocation via game theory for a small-cell video caching system," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2115–2129, Aug 2016.

[3] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for UAV-enabled multiple access," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[4] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[5] L. A. Wolsey, *Integer Programming.* Wiley, New York, 1998.

[6] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, Apr. 2017.

[7] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods (Partial Solutions Manual).* Athena Scientific, 2015.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, U.K.: Cambridge Univ. Press, 2004.

[9] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Jun. 2017.