Optimal Task Offloading Scheduling for Energy Efficient D2D Cooperative Computing

Qijie Lin, Feng Wang, *Member, IEEE*, and Jie Xu, *Member, IEEE*

***Abstract*— This letter investigates energy-efficient computation offloading designs for device-to-device (D2D) cooperative computing between the two users, in which each user has time-varying computation task arrivals. In this setup, the two users can dynamically exchange the computation loads via D2D offloading for reducing the overall energy consumption. In particular, we minimize the weighted sum-energy consumption of both users over a finite time horizon, by jointly optimizing their local computing and task exchange (offloading) decisions over time, subject to the newly introduced task causality and completion constraints. By applying the convex optimization technique, we obtain the well-structured optimal solution to this problem. Numerical results show that by enabling bidirectional computation sharing between the users, the proposed D2D cooperative computing design significantly reduces the system energy consumption, as compared with other benchmark schemes.**

***Index Terms*— Mobile edge computing (MEC), cooperative computing, device-to-device offloading, convex optimization.**

I. Introduction

RECENT advancements in artificial intelligence (AI) and Internet-of-things (IoT) have enabled various low-latency and computation-heavy applications (e.g., augmented reality and tele-surgery) among massive wireless devices (e.g., smart phones, wearable devices, and IoT sensors/acutators). Towards this end, mobile edge computing (MEC) [1]–[4] has emerged as an efficient solution to provide end users with cloud-like computation capability at the network edge (e.g., access points (APs) and base stations (BSs)), which has attracted growing research interests from both academia and industry.

In addition to deploying dedicated MEC servers at APs and BSs, enabling massive wireless devices for cooperative computing is another promising technique to greatly enhance the computation performance, by exploiting abundant computation resources distributed at end devices. By allowing direct communication between wireless devices without infrastructures, device-to-device (D2D) communications achieve the significant gains in increasing area spectral efficiency, improving network coverage, and decreasing end users' power consumption [5], [6]. Such D2D communications can be realized in a network-assisted or ad hoc manner over licensed/unlicensed spectrum. Building upon D2D communications, D2D based cooperative computing has been recently proposed to enable wireless devices to share/contribute their unused computation resources among themselves for improving the overall computation performance or saving energy, where some active-computing devices can directly offload computation tasks to nearby idle devices via D2D links [7]–[11]. Note that computation tasks usually arrive at wireless devices in a time/spatial-varying fashion in practice. Therefore, it is highly possible that some users may be deficit in computation resources but surplus in communication resources, while other users may have considerable computation resource unused. In this case, the D2D cooperative computing allows computation-deficit users to use their communication resources (via D2D offloading) for trading for remote computation resources at nearby users, thereby enhancing the computation performance.

In the literature, there have been a handful of prior works investigating such D2D user cooperative computing, which can be implemented either assisted by cellular infrastructures (see, e.g., [7]–[9]) or without them (see, e.g., [10], [11]). However, these prior works mainly focused on the one-shot optimization under static computation tasks at users [7]–[10], or considered uni-directional task offloading with one task-heavy user seeking for assistances from nearby idle users [11]. Under practical time-varying task arrivals, how to design bidirectional D2D offloading over time for gaining mutual benefits among users is an interesting problem that has not been well addressed yet.

Motivated by the above works, in this letter we consider a basic two-user cooperative computing system over a finite time horizon consisting of multiple slots. The two-user model is actually a fundamental building block for more general scenarios with more than two peer users, thus helping characterize the fundamental design insights. Differently from the prior work [7] considering an infrastructure-assisted user cooperation in both communication and computation under given tasks, this letter pursues an infrastructure-free D2D cooperative computing design by additionally considering dynamic task arrivals over time. Suppose that at each user the computation tasks arrive dynamically at the beginning of each slot and should be successfully executed before the end of the last slot of the horizon. We consider a partial offloading operation, such that each user can arbitrarily partition its tasks into two parts for local computing and D2D offloading to the other user per slot, respectively. The main contribution of this letter is summarized as follows. Under this two-user system setup, we first develop a new design framework to

Manuscript received July 8, 2019; accepted July 24, 2019. Date of publication July 29, 2019; date of current version October 9, 2019. This work was supported in part by the National Science Foundation of China (No. 61871137), the Guangdong Province Key Area R&D Program (No. 2018B030338001 and No. 2019B010119001), the Natural Science Foundation of Guangdong Province (No. 2018A030310537), and the National Key Program of China (No. SQ2018YFB180012). The associate editor coordinating the review of this letter and approving it for publication was K. Rajawat. *(Corresponding author: Feng Wang.)*

The authors are with the School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: tingyu07j@gmail.com; fengwang13@gdut.edu.cn; jiexu@gdut.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2019.2931719

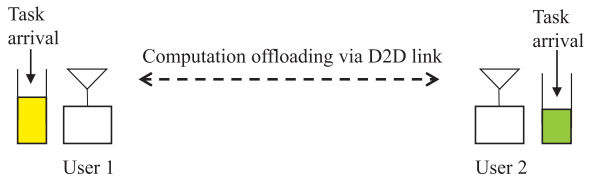

Fig. 1. System model.

minimize the two users' weighted sum-energy consumption over the horizon subject to the task causality and completion constraints, by jointly optimizing their local computing and task exchange (offloading) decisions over different slots. Then, to characterize the fundamental performance upper bound, we consider the offline optimization by assuming perfect knowledge of channel state information and task arrivals is known *a-priori*. By using the convex optimization techniques, we obtain the optimal offline solution in a semi-closed form. It is shown that both users' local computation rates monotonically increase over time, and the offloading decision at both users depends on their computation rates. Numerical results are finally provided to show that by enabling bidirectional computation sharing between the two users, the proposed D2D cooperative computing design significantly reduces the system energy consumption, as compared with other benchmark schemes.

II. System Model and Problem Formulation

As shown in Fig. 1, we consider a basic two-user D2D cooperative computing system with dynamic task arrivals. We focus on a finite time horizon of duration $T > 0$, which is divided into a set $\mathcal{N} \triangleq \{1, \ldots, N\}$ of N slots each with identical duration $\tau = T/N$. Let $L_{i,n}$ denote the number of task input-bits randomly arrived at user $i \in \{1, 2\}$ at the beginning of slot $n \in \mathcal{N}$. We consider a partition offloading operation, such that each user can execute its tasks by arbitrarily partitioning its tasks into two parts for local computing and offloading to the other user, respectively. Furthermore, we let $\omega_{i,n}$ and $\ell_{i,n}$ denote the number of task input-bits for local computing at user i and offloading from user i to user $\bar{i} \in \{1, 2\} \setminus \{i\}$ at each slot $n \in \mathcal{N}$, respectively.

First, we consider the task causality constraints, such that the cumulative number of task input-bits for user i to execute (i.e., $\sum_{j=1}^{n}(\ell_{i,j} + \omega_{i,j})$ via user i's offloading and local computing) at each slot $n \in \mathcal{N} \setminus \{N\}$ cannot exceed that cumulatively arrived at user i (i.e., $\sum_{j=1}^{n}(L_{i,j} + \ell_{\bar{i},j})$). Notice that the term $\sum_{j=1}^{n}(L_{i,j} + \ell_{\bar{i},j})$ represents the cumulative number of task input-bits arrived at user i until slot n including the dynamically arrived tasks $\{L_{i,j}\}_{j=1}^{n}$ and user $\bar{i}$'s offloaded ones $\{\ell_{\bar{i},j}\}_{j=1}^{n}$. As a result, we have, $\forall i \in \{1, 2\}$,

$$\sum_{j=1}^{n}(L_{i,j} + \ell_{\bar{i},j} - \ell_{i,j} - \omega_{i,j}) \geq 0, \quad \forall n \in \mathcal{N} \setminus \{N\}. \quad (1)$$

In addition, we impose a "hard" completion deadline for the two users to execute their tasks, i.e., all per-slot randomly arrived tasks at the users should be successfully completed before the end of this horizon. Therefore, our proposed design will not introduce more delay beyond the horizon. Correspondingly, the task completion constraints at the users are

$$\sum_{j=1}^{N}(L_{i,j} + \ell_{\bar{i},j} - \ell_{i,j} - \omega_{i,j}) = 0, \quad \forall i \in \{1, 2\}. \quad (2)$$

Next, we consider the computation energy consumption. Let γ_i and C_i denote the effective capacitance coefficient and the number of CPU cycles required for computing one task input-bit for user i, respectively. Therefore, the energy consumption for local computing at user $i \in \{1, 2\}$ at slot $n \in \mathcal{N}$ is given as $E_{i,n}^{\text{comp}} = C_i \omega_{i,n} \gamma_i \left(\frac{C_i \omega_{i,n}}{\tau}\right)^2 = \frac{\gamma_i C_i^3 \omega_{i,n}^3}{\tau^2}$.

Then, we consider the D2D offloading energy consumption. As an initial investigation, in this letter we consider a block fading channel model for D2D offloading between the two users, in which the channel power gain for offloading is assumed to remain unchanged over the horizon but may become different over different horizons. As in [9], the D2D link for offloading can be established in a network-assisted manner. Furthermore, we assume the channel reciprocity holds for offloading between the two users. We denote $h > 0$ as the channel power gain for offloading between the users, which can be obtained via pilot-based training methods [6]. It is assumed that h is perfectly known at both users. At each slot $n \in \mathcal{N}$, the achievable transmission rate (in bits/second) for user i to offload tasks to user $\bar{i}$ is $r_{i,n} = B \log_2(1 + \frac{p_{i,n} h}{\sigma_0^2})$, where $p_{i,n}$ represents user i's transmission power at slot n, B the bandwidth, and σ_0^2 the noise power at each user. In order to offload the number of task input-bits $\ell_{i,n}$ from user i to user $\bar{i}$, it is clear that $r_{i,n}\tau = \ell_{i,n}, \forall i \in \{1, 2\}, n \in \mathcal{N}$. Accordingly, the energy consumption for task offloading at user $i \in \{1, 2\}$ at slot $n \in \mathcal{N}$ is given by $E_{i,n}^{\text{off}} = \tau p_{i,n} = \frac{\sigma_0^2 \tau}{h}(2^{\frac{\ell_{i,n}}{\tau B}} - 1)$.

In this letter, our objective is to minimize the weighted sum-energy consumption of both users,[1] by jointly optimizing their local computing and task offloading decisions over time, subject to both the task causality constraints in (1) and the task completion constraints in (2). Let $\alpha_i > 0$ denote the energy weight for user $i \in \{1, 2\}$, which is used to characterize user i's priority or preference in system designs. The weighted sum energy minimization problem is thus formulated as

$$(\text{P1}): \min_{\{\ell_{i,n}, \omega_{i,n}\}} \sum_{i=1}^{2}\sum_{n=1}^{N} \alpha_i \left(\frac{\sigma_0^2 \tau}{h}(2^{\frac{\ell_{i,n}}{\tau B}} - 1) + \frac{\gamma_i C_i^3 \omega_{i,n}^3}{\tau^2}\right)$$
$$\text{s.t. (1) and (2)}$$
$$\ell_{i,n} \geq 0, \quad \omega_{i,n} \geq 0, \ \forall i \in \{1, 2\}, \ n \in \mathcal{N}.$$

Note that problem (P1) is technically quite challenging to solve, due to the constraints in (1) and (2) that are imposed to account for the time-dynamics in task arrivals. In order to characterize the fundamental performance limit, we consider the offline optimization by assuming the perfect knowledge of task arrivals $\{L_{i,j}\}_{j=1}^{N}$ is known *a-priori*. In this case, problem

[1]Note that the energy consumption at each user generally consists of two parts: the flexible energy (including transmission for offloading and CPU execution for local computing) and the fixed circuit/cooling energy consumption. Since the fixed part can be safely modeled as a constant, in this letter we focus on optimizing the flexible energy part for the users' weighted sum energy minimization without loss of optimality.

(P1) is a convex optimization problem that can be solved via standard convex optimization techniques [12].

III. Optimal Offline Solution to Problem (P1)

Since the offline problem (P1) is convex and satisfies the Slater's condition, the optimal solution to (P1) can be obtained by the Karush-Kuhn-Tucker (KKT) conditions [12]. Let $\lambda_{i,n} \geq 0$, $\lambda_{i,N} \in \mathbb{R}$, $\mu_{i,j} \geq 0$, and $\nu_{i,j} \geq 0$ denote the Lagrange multipliers associated with the (i,n)-th constraint in (1), the i-th constraint in (2), $\omega_{i,j} \geq 0$, and $\ell_{i,j} \geq 0$, $\forall i \in \{1,2\}, n \in \mathcal{N} \setminus \{N\}, j \in \mathcal{N}$, respectively. Then the optimal primal solution of $\{\omega^*_{i,n}, \ell^*_{i,n}, \forall i \in \{1,2\}, n \in \mathcal{N}\}$ to (P1) and the optimal dual solution of $\{\lambda^*_{i,n}, \mu^*_{i,n}, \nu^*_{i,n}, \forall i \in \{1,2\}, n \in \mathcal{N}\}$ must satisfy the following KKT conditions:

$$\omega^*_{i,n} \geq 0, \quad \ell^*_{i,n} \geq 0, \quad \forall n \in \mathcal{N} \tag{3a}$$

$$\sum_{j=1}^{n}(L_{i,j} + \ell^*_{\bar{i},j} - \ell^*_{i,j} - \omega^*_{i,j}) \geq 0, \quad \forall n \in \mathcal{N} \setminus \{N\} \tag{3b}$$

$$\sum_{j=1}^{N}(L_{i,j} + \ell^*_{\bar{i},j} - \ell^*_{i,j} - \omega^*_{i,j}) = 0 \tag{3c}$$

$$\lambda^*_{i,j} \geq 0, \ \ \forall j \in \mathcal{N} \setminus \{N\}, \ \mu^*_{i,n} \geq 0, \ \nu^*_{i,n} \geq 0, \ \forall n \in \mathcal{N} \tag{3d}$$

$$\mu^*_{i,n}\omega^*_{i,n} = 0, \quad \nu^*_{i,n}\ell^*_{i,n} = 0, \ \forall n \in \mathcal{N} \tag{3e}$$

$$\lambda^*_{i,n}\Big(\sum_{j=1}^{n}(L_{i,j} + \ell^*_{\bar{i},j} - \ell^*_{i,j} - \omega^*_{i,j})\Big) = 0, \quad \forall n \in \mathcal{N} \tag{3f}$$

$$\frac{3\alpha_i\gamma_i C_i^3(\omega^*_{i,n})^2}{\tau^2} - \lambda^*_{i,N} + \sum_{j=n}^{N-1}\lambda^*_{i,j} = 0, \quad \forall n \in \mathcal{N} \tag{3g}$$

$$\frac{\alpha_i\sigma_0^2 \ln 2}{Bh} 2^{\frac{\ell^*_{i,n}}{\tau B}} - \lambda^*_{i,N} + \sum_{j=n}^{N-1}\lambda^*_{i,j} + \lambda^*_{\bar{i},N} - \sum_{j=n}^{N-1}\lambda^*_{\bar{i},j} = 0 \quad \forall n \in \mathcal{N}, \tag{3h}$$

where $i \in \{1,2\}$. Note that (3a–c) denote the primal feasible conditions, (3d) denotes the dual feasible conditions, (3e–f) denote the complementary slackness conditions, and (3g–h) denote the first-order derivative optimality conditions.

Proposition 1: The optimal number of task input-bits $\{\omega^*_{i,n}\}$ for local computing and $\{\ell^*_{i,n}\}$ for offloading to problem (P1) are given as[2]

$$\omega^*_{i,n} = \tau\sqrt{\left[\frac{\lambda^*_{i,N} - \sum_{j=n}^{N-1}\lambda^*_{i,j}}{3\alpha_i\gamma_i C_i^3}\right]^+} \tag{4a}$$

$$\ell^*_{i,n} = \tau B \log_2\left(\max\left[1, \frac{Bh\big(\lambda^*_{i,N} - \sum_{j=n}^{N-1}\lambda^*_{i,j} - \lambda^*_{\bar{i},N} + \sum_{j=n}^{N-1}\lambda^*_{\bar{i},j}\big)}{\alpha_i\sigma_0^2 \ln 2}\right]\right), \tag{4b}$$

respectively, where $i \in \{1,2\}$, $n \in \mathcal{N}$, and $[x]^+ \triangleq \max(x,0)$.

Proof: The optimal solution of $\{\omega^*_{i,n}\}$ and $\{\ell^*_{i,n}\}$ can be readily obtained based on the first-order derivative conditions (3g) and (3h) together with the primal and dual feasible conditions in (3a) and (3d), and the complementary slackness conditions in (3f), respectively. ■

[2]The optimal dual variables $\{\lambda^*_{i,n}, \mu^*_{i,n}, \nu^*_{i,n}\}$ can be efficiently obtained by solving the dual problem of (P1) via the subgradient based algorithms (e.g., the ellipsoid method) [13], for which the details are omitted for brevity.

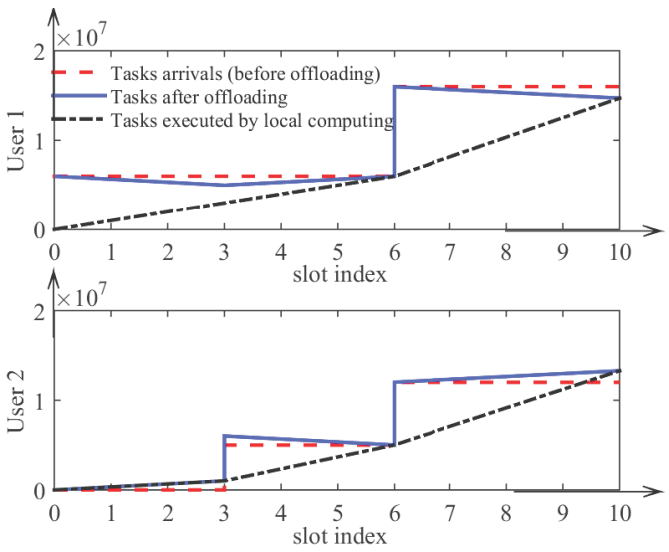

Fig. 2. Illustrations of the cumulative number of task input-bits for local computing and offloading at the two users under the proposed joint cooperative computation and offloading policy, where the system parameters are set same as those in Fig. 4 in Section IV.

Proposition 2: The optimal solution $\{\omega^*_{i,n}, \ell^*_{i,n}\}$ to problem (P1) has the following properties:

- The optimal number of task-input bits for local computing at users is monotonically increasing over time, i.e., $\omega^*_{i,1} \leq \omega^*_{i,2} \leq \cdots \leq \omega^*_{i,N}$, $\forall i \in \{1,2\}$;
- User i shall increase its local computing rate at slot $n \in \mathcal{N} \setminus \{1\}$ (i.e., $\omega^*_{i,n} > \omega^*_{i,n-1}$), only after its task buffer at that slot is empty (i.e., $\sum_{j=1}^{n-1}(L_{i,j} + \ell_{\bar{i},j} - \ell_{i,j} - \omega_{i,j}) = 0$);
- If user i starts offloading tasks to the other user $\bar{i}$ at slot n, then its per-bit energy consumption for local computing must be greater than that at user $\bar{i}$ at that slot, i.e., $\frac{\alpha_i\gamma_i C_i^3(\omega^*_{i,n})^2}{\tau^2} > \frac{\alpha_{\bar{i}}\gamma_{\bar{i}} C_{\bar{i}}^3(\omega^*_{\bar{i},n})^2}{\tau^2}$.

Proof: The first property follows directly based on (4a), together with the fact that $\lambda^*_{i,n} \geq 0$, $\forall n \in \mathcal{N}$. Next, it follows from (4a) that at each slot $n \in \mathcal{N}\setminus\{1\}$, we have $\omega^*_{i,n} > \omega^*_{i,n-1}$, only when $\lambda^*_{i,n-1} > 0$. Based on the complementary slackness conditions in (3f), it must hold that $\sum_{j=1}^{n-1}(L_{i,j} + \ell^*_{\bar{i},j} - \ell^*_{i,j} - \omega^*_{i,j}) = 0$. Therefore, the second property is verified.

Furthermore, we recast (4a) as $[\lambda^*_{i,N} - \sum_{j=n}^{N-1}\lambda^*_{i,j}]^+ = \frac{3\alpha_i\gamma_i C_i^3(\omega^*_{i,n})^2}{\tau^2}$, $\forall i \in \{1,2\}$. By substituting it into (4b), the optimal task allocation for offloading is expressed as $\ell^*_{i,n} = \tau B \log_2\big(\max(1, \frac{3Bh}{\alpha_i\sigma_0^2 \ln 2}(\frac{\alpha_i\gamma_i C_i^3(\omega^*_{i,n})^2}{\tau^2} - \frac{\alpha_{\bar{i}}\gamma_{\bar{i}} C_{\bar{i}}^3(\omega^*_{\bar{i},n})^2}{\tau^2}))\big)$. In this case, we have $\frac{\alpha_i\gamma_i C_i^3(\omega^*_{i,n})^2}{\tau^2} > \frac{\alpha_{\bar{i}}\gamma_{\bar{i}} C_{\bar{i}}^3(\omega^*_{\bar{i},n})^2}{\tau^2}$ if $\ell^*_{i,n} > 0$, which proves the third property. We now complete the proof. ■

Example 3: For illustration, Fig. 2 shows the cumulative number of task input-bits at the two users under the proposed joint cooperative computation and offloading design, including that arrived at each user i (i.e., $\sum_{j=1}^{n} L_{i,j}$), that after offloading (i.e., $\sum_{j=1}^{n}(L_{i,j} + \ell_{\bar{i},j} - \ell_{i,j})$), and that locally executed

by user i (i.e., $\sum_{j=1}^{n}\omega_{i,j}$). First, it is observed that the slope of the "tasks-executed-by-local-computing" curve at each user increases monotonically over time, which corroborates the monotonically increasing property of the number of locally executed task input-bits $\omega_{i,n}^*$'s, as shown in the first property of Proposition 2. Second, it is observed that the slope of the "tasks-executed-by-local-computing" curve increases only after the "tasks-after-offloading" and "tasks-executed-by-local-computing" curves coincide, or equivalently, the task buffer at that user is empty (c.f. the second property of Proposition 2). Third, it is observed that in the slots without task arrivals, the number of task input-bits after offloading at each user increases when the slope of the "tasks-executed-by-local-computing" curve at that user is lower than that at the other user. This is consistent with the third property of Proposition 2.

To gain more insights, we now consider the following two special cases when computation tasks arrive only at the beginning of the horizon and only at one user, respectively.

1) Case With Tasks Arrivals Only at the Beginning of the Horizon: In this case, we have $N = 1$ and $\tau = T$, and the optimal solution is given as

$$\omega_{i,1}^* = T\sqrt{\left[\frac{\lambda_{i,1}^*}{3\alpha_i\gamma_i C_i^3}\right]^+} \tag{5a}$$

$$\ell_{i,1}^* = TB\log_2\left(\max\left[1, \frac{Bh\left(\lambda_{i,1}^* - \lambda_{\bar{i},1}^*\right)}{\alpha_i\sigma_0^2\ln 2}\right]\right), \tag{5b}$$

where $i \in \{1,2\}$ and the optimal Lagrange multipliers $\lambda_{1,1}^*$ and $\lambda_{2,1}^*$ can be obtained via a bisection search based on the two equations in (3c), together some simple manipulations. It is evident from (5a) and (5b) that the task offloading is always implemented from user i to user $\bar{i}$, since user i has a higher local computing rate than user $\bar{i}$, which is consistent with the third property of Proposition 2.

2) Case With Tasks Arrivals Only at User 1: In this case, only user 1 has dynamic task arrivals and user 2 acts as a helper that is willing to share its computation resources to reduce the weighted sum-energy consumption. Accordingly, only user 1 offloads tasks to user 2, i.e., $\ell_{2,n}^* = 0$, $\forall n \in \mathcal{N}$. It can also be verified that $\ell_{1,n}^* = \omega_{2,n}^*$, $\forall n \in \mathcal{N}$. Based on the above observations and similarly as for solving problem (P1), the optimal solution in this case is expressed as

$$\omega_{1,n}^* = \tau\sqrt{\left[\frac{\lambda_{1,N}^* - \sum_{j=n}^{N-1}\lambda_{1,j}^*}{3\alpha_1\gamma_1\, C_1^3}\right]^+}$$

$$\ell_{1,n}^* = \tau B\log_2\left(\max\left[1, \frac{Bh(\lambda_{1,N}^* - \sum_{j=n}^{N-1}\lambda_{1,j}^* - \frac{3\alpha_2\gamma_2 C_2^3 \ell_{1,n}^{*2}}{\tau^2})}{\alpha_1\sigma_0^2\ln 2}\right]\right),$$

where $n \in \mathcal{N}$. Note that $\ell_{1,n}^*$ and $\omega_{1,n}^*$ are only related to the term $\lambda_{1,N}^* - \sum_{j=n}^{N-1}\lambda_{1,j}^*$. Therefore, the values of $\omega_{1,n}^*$, $\ell_{1,n}^*$, and $\omega_{2,n}^* = \ell_{1,n}^*$ are all monotonically increasing over time. For this reason, the optimal solution of $\{\omega_{1,n}^*, \ell_{1,n}^*\}$ in this case can be efficiently obtained via a staircase-like task allocation algorithm, similarly as the staircase power allocation policy for energy-harvesting based wireless communications [14], for which the details are omitted for brevity.

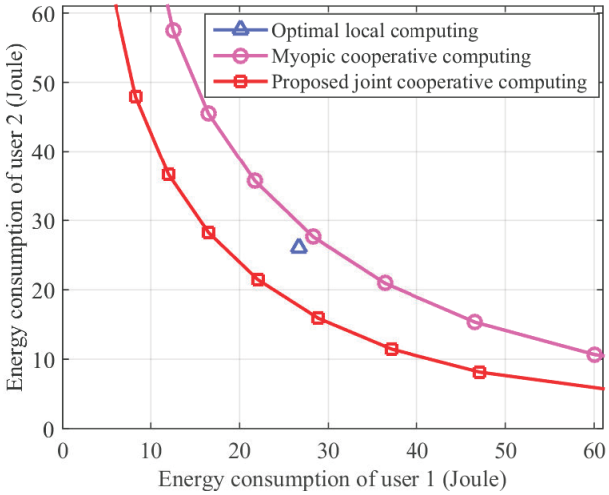

Fig. 3. The energy consumption tradeoff between the two users.

IV. Numerical Results

In this section, we provide numerical results to demonstrate the performance of our proposed joint cooperative computing design. For comparison, we consider the following two benchmark schemes.

- *Optimal Local Computing Only:* In this scheme, all tasks at each user are executed only by its own local computing. This corresponds to a special case of problem (P1) with $\ell_{i,n} = 0$, $\forall i \in \{1,2\}$, $n \in \mathcal{N}$, for which optimal solution of $\{\omega_{i,n}\}$ admits a similar structure as (4a).
- *Myopic Cooperation Computing:* In this scheme, the users shall complete the task execution for their arrived tasks at each slot independently. At each slot, the two users can find their optimal local computing and offloading decisions similarly as in (5a) and (5b), respectively.

In the simulations, we assume that the number of each user's task input-bits at each slot is independent and identically distributed with a uniform distribution within the range $[0, L_i^{\max}]$, where $L_i^{\max}$ denotes user i's maximal number of arrived task input-bits. The system parameters are set as follows. The length of the horizon is $T = 0.2$ seconds, and the number of time slots is $N = 10$. We set the receiver noise power as $\sigma_0^2 = 10^{-9}$ Watt, the bandwidth $B = 1$ MHz and the channel power gain $h = 10^{-6}$ for D2D offloading, where the distance between the users is around 10 meters. Suppose that the users are with the identical switch capacitance coefficient $\gamma_1 = \gamma_2 = 10^{-28}$, and the required number of CPU cycles per bit is set as $C_1 = C_2 = 500$ cycles/bit.

Fig. 3 shows the energy consumption tradeoff between the two users, where $L_1^{\max} = L_2^{\max} = 0.8$ Mbits with $\alpha_1 + \alpha_2 = 1$. It is observed that as α_1 increases, the energy consumption of user 1 decreases but the energy consumption of user 2 increases. As compared to the two benchmark schemes, our

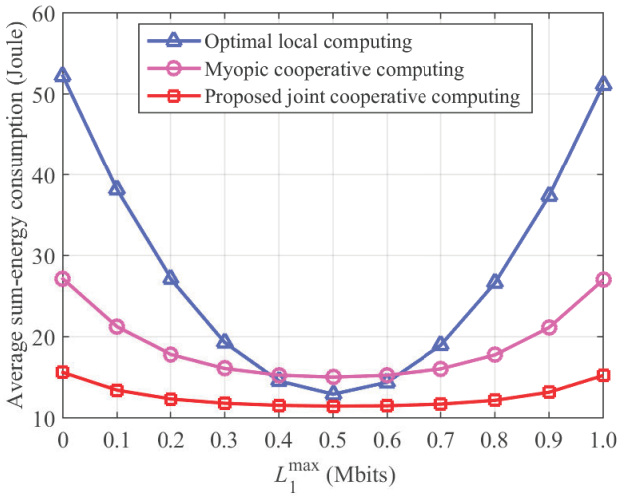

Fig. 4. Average sum-energy consumption of the users versus $L_1^{\max}$.

proposed joint cooperative computing design is observed to achieve significantly lower energy consumption at both users.

Fig. 4 shows the average sum-energy consumption of the users versus $L_1^{\max}$, where $L_1^{\max} + L_2^{\max} = 1$ Mbits and the energy weights are set as $\alpha_1 = \alpha_2 = \frac{1}{2}$. It is observed that our proposed joint cooperative computing design outperforms the benchmark schemes in energy saving. Interestingly, the myopic cooperative computing scheme is observed to outperform the optimal local computing one, when the difference of task input-bits between users becomes large (e.g., $L_1^{\max} < 0.4$ Mbits or $L_1^{\max} > 0.6$ Mbits). This further illustrates the merit of task offloading for energy saving.

V. Conclusion

This letter investigated an energy-efficient D2D cooperative computing design to minimize the two users' weighted sum-energy consumption within a given finite time horizon. Under the task causality and task completion constraints, we jointly optimized the task allocation for the local computing and task exchange (offloading) over time. Numerical results demonstrated the merit of the proposed D2D cooperative computing design in energy saving, as compared with other benchmark schemes. It is our hope that this work can provide some inspirations to explore dynamic D2D task sharing for improving the computation performance of massive devices in the IoT era. In this work, we considered energy minimization under given computation latency requirements, while our design can also be extended to characterize the interesting energy-delay tradeoff by adjusting the computation latency requirements. Furthermore, how to analyze the individual latency performance for different computation tasks at both users is also an interesting direction worthy of investigation.

References

[1] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Mar. 2017.

[2] 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, Aug. 2017.

[3] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[4] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Mar. 2019.

[5] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, "Device-to-device communications in cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 49–55, Apr. 2014.

[6] A. Asadi, Q. Wing, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, Apr. 2014.

[7] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4188–4200, Jun. 2019.

[8] D. Wu, F. Wang, X. Cao, and J. Xu, "Wireless powered user cooperative computation in mobile edge computing systems," in *Proc. IEEE GLOBECOM Workshops*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

[9] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[10] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint task assignment and wireless resource allocation for cooperative mobile-edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4193–4207, Jun. 2019.

[11] C. You and K. Huang, "Exploiting non-causal CPU-state information for energy-efficient mobile cooperative computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4104–4117, Jun. 2018.

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

[13] S. Boyd. (2013, Sep.). *Convex Optimization II*. Stanford University. [Online]. Available: http://www.stanford.edu/class/ee364b/lectures.html

[14] C. K. Ho and R. Zhang, "Optimal energy allocation for wireless communications with energy harvesting constraints," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4808–4818, Sep. 2012.