Coalitional Games for Computation Offloading in NOMA-Enabled Multi-Access Edge Computing

Article *in* IEEE Transactions on Vehicular Technology · November 2019

DOI: 10.1109/TVT.2019.2956224

Coalitional Games for Computation Offloading in NOMA-Enabled Multi-Access Edge Computing

Quoc-Viet Pham, *Member, IEEE,* Hoang T. Nguyen, Zhu Han, *Fellow Member, IEEE,* and Won-Joo Hwang, *Senior Member, IEEE*

***Abstract*—Multi-access edge computing (MEC) and nonorthogonal multiple access (NOMA) are two enabling technologies in the 5G network and beyond. MEC admits user equipments (UEs) running many more compute-intensive applications by providing computing capabilities at the network edge and within radio access networks, while NOMA enables multiple UEs to share the same resource block, thus leveraging considerable advantages such as greater spectral efficiency and a larger number of supported UEs. The state-of-the-art showed that the combination of NOMA and MEC can lower the energy consumption and/or overall latency; however, they mostly focused on single-carrier NOMA. In this paper, we investigate the computation offloading problem in multi-carrier NOMA enabled MEC systems and solve it from the cooperative game theory viewpoint using coalition formation game. Particularly, UEs are considered as game players and subcarriers are regarded as coalitions that can be used for computation offloading of multiple UEs. Based on the introduced coalition formation game, we develop a low-complexity algorithm with convergence guarantee to achieve the Nash-stable solution. Numerical results are provided to validate the effectiveness of the proposed coalition game based algorithm as well as its comparison with three baseline schemes.**

***Index Terms*—Coalitional Game Theory, Computation Offloading, Multi-Access Edge Computing (MEC), Non-Orthogonal Multiple Access (NOMA), Resource Allocation.**

I. Introduction

With the development of Internet of Things (IoT) technologies and popularity of mobile devices (e.g., wearable computing devices and virtual reality glass), the global mobile traffic is expected to increase at an exponential rate [1]. An Ericsson report in [2] reveals that mobile traffic will increase twelvefold between 2016 and 2022. In response to the unprecedented amount of data traffic and the emergence of new resource-demanding applications, for example, 3D gaming, data analytics, and mobile blockchain, edge computing has become a very realistic solution. To meet such high computing demands at the edge of the network (also known as network edge, i.e., near the end users), in late 2014 European Telecommunications Standards Institute (ETSI) introduced *multi-access edge computing* (MEC) to provide "IT-based services and cloud computing capabilities within the radio access network in the vicinity of UEs" [3]. In other words, the main purpose of MEC is to move mobile communication, computing, caching, and control to the network edge. Compared with cloud computing, MEC takes considerable advantages of reducing the latency, offering energy savings for UEs, achieving higher reliability [4]–[6]. Therefore, MEC has been considered as one of the enabling technologies for ultra-reliable and low-latency (URLLC) communications [6], [7].

NOMA, short for non-orthogonal multiple access, has been emerged as a key component of radio access techniques in the 5G network [8]. The fundamental idea of NOMA is using superposition coding to superpose signals from various users at the transmitter side and successive interference cancellation to decode the intended signals at the receiver side. NOMA has the potential to accommodate more UEs than the number of available subcarriers, which can improve wireless communication with multiple potentials, including massive connectivity, lower latency, higher spectral efficiency, and relaxed channel feedback [9]. First, theoretically NOMA can serve multiple UEs in an available resource, and thus NOMA is a key technology for massive connectivities (e.g., Internet of Things). Next, grant-free access and flexible scheduling can be enabled by NOMA so that more UEs can be served simultaneously, thus reducing the waiting latency. Third, NOMA demonstrates the fairness provisioning and spectral efficiency enhancement over conventional orthogonal multiple access (OMA) technologies. Such better performance is achieved since NOMA users can utilize all the subcarrier channels, whereas OMA users can only enjoy a small fraction of the entire spectrum [5], [8]. Lastly, power-domain NOMA can relax the requirement of channel feedback since accurate CSI is only used for power allocation [10].

Motivated by the superiority of NOMA over OMA [9], [10], the performance of MEC systems can be much enhanced by NOMA, as compared to that of conventional OMA-based MEC approaches. Let us give an example with two UEs, as shown in Fig. 1, to illustrate this point. Here, UE 1 needs to send emergent small-size packets to the eNB with a low required data rate, but within a very short time duration, and UE 2 needs to migrate a computation-heavy and delay-tolerant task to the MEC server for remote processing. Basically, OMA-based approaches serve these two UEs with two orthogonal subcarriers/time slots, which may result in

Quoc-Viet Pham and Won-Joo Hwang are with Department of Biomedical Convergence Engineering, Pusan National University, 2 Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, 46241 Republic of Korea. Hoang T. Nguyen is with Department of Information and Communication System, 197, Inje-ro, Gimhae-si, Gyeongsangnam-do, 50834 Republic of Korea. Zhu Han is with Department of Electrical and Computer Engineering, University of Houston, 4800 Calhoun Rd, Houston, TX 77004, USA. E-mail: {vietpq90@gmail.com, hoangnt192@gmail.com, zhan2@uh.edu, won-joo.hwang@gmail.com}.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (No.NRF-2019R1C1C1006143). Quoc-Viet Pham is the corresponding author.

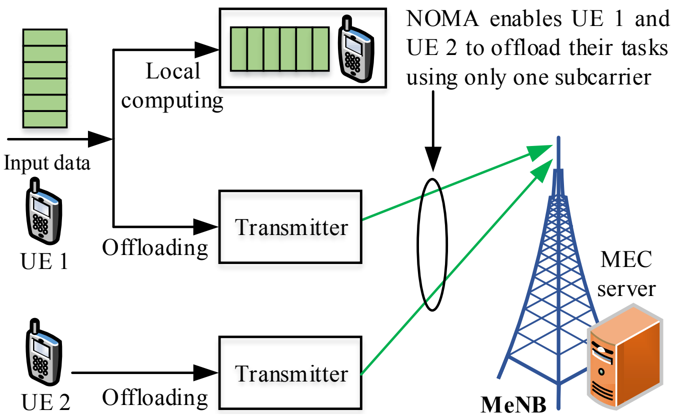

Fig. 1: Example of MEC computation offloading using NOMA.

the low efficiency of resource allocation and abysmal lack of resources in the case of massive connectivities. NOMA can overcome these drawbacks since two NOMA UEs can be served using the same spectrum resource. As can be seen, Fig. 1 shows that when two UEs select remote computing, NOMA enables them to offload their computation tasks using only one radio resource block (e.g., orthogonal frequency division multiplexing (OFDM) subcarrier). Now, we suppose that UE 2 has a big-data and latency-tolerant computation task and two orthogonal subcarriers are available. In this case, a feasible solution is utilizing NOMA in the first subcarrier for computation offloading of UE 1 and a part of UE 2 while utilizing OMA in the second subcarrier for computation offloading of the remaining workload of UE 2.

The exploitation of NOMA in MEC systems has been recently investigated in many recent works. The energy minimization problem was studied in [11]–[15] and the latency minimization was considered in [16]–[21], and joint completion time and energy consumption was minimized in [22]. The direct observation from existing papers is that the combination of NOMA and MEC can produce substantial reductions on energy consumption and/or overall latency, which depends largely on the problem formulation. However, most of the state-of-the-art studies focused on single-carrier NOMA. Concretely, optimization and performance analysis of NOMA-enabled MEC are performed under a typical network setting, where the time duration is slotted into orthogonal frames and UEs in the same frame utilize NOMA for their computation offloading. For example, a hybrid-NOMA approach with two UEs was studied in [16], where NOMA is adopted in the first time slot while the UE with latency-tolerant applications is allowed to offload its remaining workload during another dedicated time slot. The authors in [14], [15], [22] considered TDMA for UEs in different NOMA groups, and jointly optimized the task offloading ratio and time allocation for different NOMA groups so as to minimize the energy consumption. We are not aware of any work pertained to MEC with multi-carrier (MC) NOMA.

In this work, we focus on a cooperative scheme among UEs in the MC NOMA-based MEC system. Due to the prohibitive time-complexity of optimal solution approaches and the need for network scalability, we address the computation offloading problem in MC NOMA-enabled MEC from the game theory perspective. Motivated by the advantages of coalitional games in addressing various problems in wireless communications and networking, e.g., D2D communication [23], [24] and cognitive radio [25], we propose in this work a coalition formation game model. In more detail, coalitional games concern a number of players, i.e., UEs in the context of this paper, who cooperatively form coalitions in order to optimize the network performance in terms of computation overhead and obtain their offloading decisions in a distributed manner. Based on the introduced coalition formation game, we propose an algorithm with low computational complexity and convergence guarantee to achieve the Nash-stable solution.

To the extent of our knowledge, this paper is the first trial that addresses computation offloading in MC-NOMA enabled MEC systems using coalitional game theory. In this regard, our main contributions are summarized as follows.

- Different from the state-of-the-art, we consider MC NOMA, a general case of NOMA, where there are multiple groups. UEs within each group are assigned the same subcarrier and are allowed to communicate with the MeNB/MEC server following the NOMA principle, and different orthogonal subcarriers are allocated to different groups. In the network setting, each UE needs to determine *(i)* which computation mode is selected, local processing or remote execution? and *(ii)* in the case of offloading, which subcarrier is used for the computation migration?
- We formulate a joint offloading decision and subcarrier assignment problem that aims to minimize the total computation overhead. To solve that mixed-integer programming problem, we introduce the coalition formation game, in which each subcarrier is regarded as a coalition and it can be used for computation offloading of multiple UEs. After that, we devise a distributed coalition formation game based algorithm and conduct theoretical properties of the proposed algorithm.
- Numerical simulations are conducted under various performance metrics and parameters in order to demonstrate the effectiveness and outperformance of our proposed algorithm in reducing the total computation overhead when compared with three alternative schemes.

The rest of this paper is structured as follows. We introduce the system models and formulate the optimization problem in Section II. In Section III, we describe the coalition formation game and present our proposed algorithm. We also conduct its theoretical analyses of convergence, stability, and complexity. Simulation results and comparisons of the proposed algorithm with three other schemes are presented in Section IV. Finally, Section V concludes this paper.

II. System Model and Problem Formulation

A. Network Model

As illustrated in Fig. 2, a network setting of N UEs and one MEC server, which is attached to the corresponding MeNB,

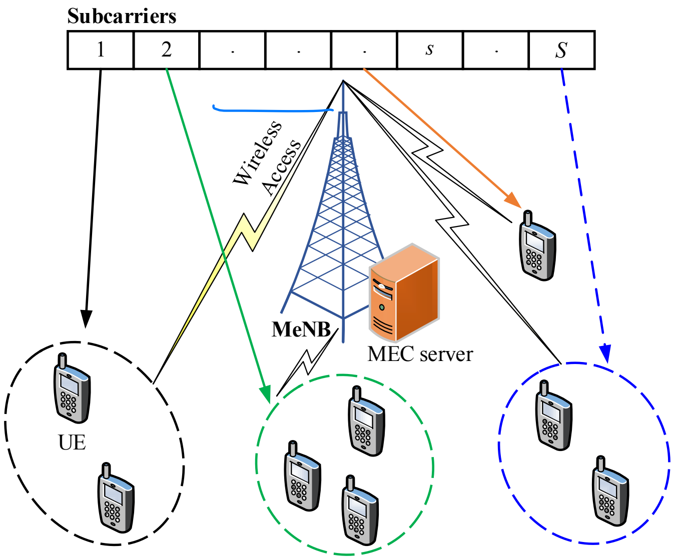

Fig. 2: Illustration of a NOMA-enabled MEC system.

is considered. In this work, we assume that the channels are quasi-static Rayleigh fading, i.e., UEs stay unchanged within each offloading period and vary independently between any two periods. Moreover, we assume that both the UEs and MeNB are equipped with one single antenna. Let $\mathcal{N} = \{1, \ldots, N\}$ and $\mathcal{S} = \{1, \ldots, S\}$ denote the set of UEs and orthogonal subcarriers, respectively. With NOMA, a subcarrier can be shared among multiple UEs, and thus the received signal of a UE at the MeNB contains not only its desired signals, but also interfering signals from co-sharing UEs. We use $\mathcal{U}_s$ to denote the set of orders of UEs sharing subcarrier s and assume that each UE can utilize at most one subcarrier to offload its computations to the MEC server. Obviously, we have $\mathcal{U}_s \cap \mathcal{U}_{s'} = \emptyset, \forall s \neq s'$, and $\bigcup_{s \in \mathcal{S}} \mathcal{U}_s = \mathcal{N}$.

B. Communication Model

Denote by $\boldsymbol{A} = \{a_{ns} | n \in \mathcal{N}, s \in \mathcal{S}\}$ the offloading decision profile, where $a_{ns} = 1$ if UE n utilizes the subcarrier s to offload its computation task and $a_{ns} = 0$ otherwise. For example, in Fig. 1 UEs 1 and 2 offload their computations using the same resource block, say the subcarrier 2, so $a_{12} = 1$ and $a_{22} = 1$. Since each UE can use at most one subcarrier for computation offloading, the offloading decisions need to satisfy the following constraint

$$\sum_{s \in \mathcal{S}} a_{ns} \leq 1, \forall n \in \mathcal{N}. \tag{1}$$

Denote by h_{ns} the uplink channel gain between UE n and the MeNB on subcarrier s. The channel gains in cluster s are sorted in the ascending order and we use the bijection $b_s(\cdot)$ to represent this order, where $b_s(j)$ denotes the position of UE j in the sorted sequence on subcarrier s. Without loss of generality, the received signals from UEs with $b_s(j) < b_s(i)$ is not decoded by UE i and thus is treated as noise. According to [26], the decoding order in the uplink NOMA follows the decreasing order of the channel gains, which is different from the downlink NOMA, where the decoding order applies in reverse.

The received signal-to-interference-plus-noise ratio (SINR) of the UE n on the subcarrier s is expressed as follows:

$$\Gamma_{ns} = \frac{p_{ns} h_{ns}}{\sum\limits_{j \in \mathcal{U}_s : b_s(j) < b_s(n)} p_{js} h_{js} + n_0}, \tag{2}$$

where p_{ns} denotes the transmit power of the UE n on the subcarrier s and n_0 denotes the noise power[1]. For the UE n on the subcarrier s, the achievable rate is $R_{ns} = B \log_2 (1 + \Gamma_{ns})$, where B is the bandwidth of an orthogonal subcarrier. Thus, the achievable rate of the UE n is given as follows:

$$\begin{aligned} R_n &= B \sum_{s \in \mathcal{S}} a_{ns} \log_2 (1 + \Gamma_{ns}) \\ &= B \sum_{s \in \mathcal{S}} a_{ns} \log_2 \left(1 + \frac{p_{ns} h_{ns}}{\sum\limits_{j \in \mathcal{U}_s : b_s(j) < b_s(n)} p_{js} h_{js} + n_0} \right). \end{aligned} \tag{3}$$

C. Computation Model

From the user perspective, computation offloading is regarded as an important use case, which enables UEs to exploit substantial computing resources at the edge so as to execute compute-intensive functionalities. Basically, a computation offloading decision can be categorized as: *local execution*, *full offloading*, and *partial offloading*. Local computing indicates that UEs have no benefit from computation offloading; accordingly, the whole task is handled locally. In full offloading, the entire task is migrated to the edge server for remote processing, while it is divided into smaller parts in partial offloading, some of which are processed locally and the remaining fraction is executed remotely. Compared with partial offloading, binary offloading (as considered in this work) is a general version, which makes the offloading decision problem difficult to solve due to the combinatorial nature of binary offloading decisions.

The computation task of each UE n is defined $I_n = \{\alpha_n, \beta_n\}$, where α_n is the data size of I_n and β_n is the required CPU cycles to accomplish I_n [28], [29]. We define x_n to characterize whether the local computing or computation offloading is preferred and selected by the UE n. If UE n decides to offload the task, $x_n = 1$; otherwise, $x_n = 0$, i.e., task I_n is executed locally by UE n. Each the UE chooses either local computing or remote execution and utilizes at most one subcarrier for computation offloading, we hold the following equality $\sum_{s \in \mathcal{S}} a_{ns} = x_n$. On the one hand, NOMA enables the sharing of one subcarrier by multiple offloading UEs. On the other hand, each UE is allowed to select between local and remote computing.

[1]The transmit power of UEs is predefined. When the joint design of computation offloading decision and power control is considered, there are some constraints on NOMA power allocation to be imposed, e.g., total transmit power for all UEs utilizing the same cluster [26] and constraint for efficient successive interference cancellation [27]. Distributed power control needs to be developed to improve the network scalability. However, since the power allocation problem is typically non-convex, one may exploit heuristic and/or meta-heuristic approaches to obtain the solution in an efficient manner. The joint optimization offloading decision and power allocation is an interesting issue and is under our study.

Regarding local computation, we denote f_n^l local as the CPU computing capability of UE n (the superscript l signifies *local*). The task I_n completion time can be given as

$$T_n^l = \frac{\beta_n}{f_n^l}. \tag{4}$$

The corresponding energy consumption E_n^l can be computed as follows:

$$E_n^l = \kappa_n \beta_n \left(f_n^l\right)^2 \tag{5}$$

where κ_n is a constant factor related to the hardware architecture [29]. The computation overhead Z_n^l is determined by the latency T_n^l and energy consumption E_n^l needed to accomplish task I_n. Similar to [28], in this paper, Z_n^l is defined below

$$Z_n^l = \lambda_n^t T_n^l + \lambda_n^e E_n^l, \tag{6}$$

which is the weighted sum of execution latency and energy consumption with λ_n^t, λ_n^e being the weighted parameters. The UEs' offloading decisions can be greatly affected by the UE weights λ_n^t, λ_n^e. Let us give an example with three UEs. UE 1 has a latency-sensitive application, thus it sets $\lambda_n^t = 1, \lambda_n^e = 0$. Due to the energy-hungry application and its low battery state, UE 2 sets $\lambda_n^e = 1, \lambda_n^t = 0$. UE 3 sets $0 < \lambda_n^t, \lambda_n^e < 1$ since both the completion time and energy consumption are needed to be considered. Alternatively, some metrics can be used instead of computation overhead, for example, computation efficiency and computation rate [30], [31].

In the case of remote execution of I_n, the completion time T_n^r (the superscript r stands for *remote*) is mainly composed of two parts[2]: uplink transmission time T_n^t and execution time T_n^e. Thus, we have $T_n^r = T_n^t + T_n^e$. Here,

$$T_n^t = \frac{\alpha_n}{R_n}, \qquad T_n^e = \frac{\beta_n}{f_n}, \tag{7}$$

and where f_n is the amount of remote computing resources allocated to UE n. We note that $f_n = 0$ when $x_n = \sum_{s \in \mathcal{S}} a_{ns} = 0$, i.e., task I_n is executed locally. In this work, we assume that the MEC server allocates each offloading UE a fixed amount of computing resources f_n. This assumption may hold when the MEC computing capability is sufficiently large, which facilitates the MEC server to provide a fixed computing service rate to multiple offloading UEs simultaneously. Moreover, the collaboration between neighbour MEC servers helps scale up their computing capabilities so as to guarantee computing resources assigned to offloading UEs. The optimization of task offloading, subcarrier assignment, and computing resource allocation is promising, which is left for future research.

The total energy consumption is composed of three parts corresponding to task offloading, remote computing, and result downloading [28]. Since we focus on the UE perspective and MeNB/MEC servers are typically powered by grid energy, we only consider energy consumed for computation offloading. Thus, we have

$$E_n^r = \frac{p_n}{\varsigma_n} T_n^t = \frac{p_n}{\varsigma_n} \frac{\alpha_n}{R_n}, \tag{8}$$

where ς_n is the UE power amplifier efficiency. Then, the computation overhead for remote execution of UE n can be calculated as $Z_n^r = \lambda_n^t T_n^r + \lambda_n^e E_n^r$.

[2]The response time of the computing result back to UEs can be ignored since the size of computing result is usually small [28], [30], [31]. When considering non-negligible computing result size, joint uplink and downlink resource optimization can be designed by adopting time-division duplexing mode in NOMA systems. This straightforward extension from our current work is an interesting topic for future work.

D. Problem Formulation

It is clear that the total computation overhead is pertained to the task offloading decisions x_n and subcarrier assignment $a_{n,s}$, $\forall s \in \mathcal{S}, n \in \mathcal{N}$. We jointly optimize the computation offloading decisions and subcarrier assignment with the aim of minimizing the total computation overhead. In this regard, we define an objective function that reflects the sum computation overhead incurred by all UEs, denoted by $Z(\boldsymbol{A})$, which can be obtained as

$$Z(\boldsymbol{A}) = \sum_{n \in \mathcal{N}} \left(x_n Z_n^r + (1 - x_n) Z_n^l\right). \tag{9}$$

The optimization problem of computation offloading in MC-NOMA enabled MEC systems can be formulated as

$$\min_{\boldsymbol{A}} \quad Z(\boldsymbol{A}) \tag{10a}$$

$$\text{s.t.} \quad a_{ns}, x_n = \{0, 1\}, \forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \tag{10b}$$

$$x_n = \sum_{s \in \mathcal{S}} a_{ns} \leq 1, \forall n \in \mathcal{N}. \tag{10c}$$

The constraints in (10b) and (10c) indicates binary offloading, i.e., a computation task can be processed using either local computing or remote execution. It is observed that the problem in (10) is a mixed-integer programming (MIP) problem due to the existence of binary and integer variables. It is worth mentioning that MIP problems are NP-hard by nature [32]. Although the optimal solution for MIP problems can be achieved using some existing methods, e.g., branch-and-bound algorithm and exhaustive search [33], the application of these approaches is usually limited due to their prohibitive time-complexity. In Section III below, we first introduce a coalition formation game for computation offloading in NOMA-enabled MEC systems and then develop a distributed algorithm to solve the optimization problem in (10) with low computational complexity.

III. Coalitional Game Approach

A. Coalition Formation Game Formulation

Using the coalition game in our investigated system, UEs form coalitions to improve the network efficiency in terms of computation overhead. Each UE n in the network is regarded as a game player, who needs to make a task offloading decision on either executing locally or migrating the computation task through a subcarrier. Since there are N UEs and S subcarriers and binary offloading is considered, and thus N UEs can form to create $S + N$ coalitions. We denote the collection of coalitions as $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_{S+N}\}$, where $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for any $i \neq j$, and $\bigcup_{j=1}^{S+N} \mathcal{F}_j = \mathcal{N}$, where the cardinality of the collection $\mathcal{F}$ measures the number of coalitions. The coalition

$\mathcal{F}_j$ with $1 \le j \le S$ refers to the set of UEs utilizing subcarrier j for computation offloading. For $\mathcal{F}_j$ with $S+1 \le j \le S+N$ is UE j that executes the computation task locally.

It is perceived that the greater the number of UEs sharing the same subcarrier for computation offloading, the greater the receiver complexity for inter-user interference cancellation. Actually, the larger number of co-sharing UEs reduces SINRs of UEs with lower channel gains due to much more severe interference, thus increasing the transmission latency and then computation overhead. In such case, there is little incentive for all UEs to utilize only one subcarrier for computation offloading while some other subcarriers with good channel conditions are available. Therefore, the formation of a grand coalition is not beneficial and an efficient coalition formation scheme can be devised to lower the total computation overhead. It is worth noting that some of the coalitions may be empty if the channel conditions of UEs on such subcarriers are not favorable and some UEs decide to handle the tasks locally. The coalition formation game with transferable utility is defined below.

Definition 1. *A coalition formation game with transferable utility for computation offloading and resource allocation in NOMA-enabled MEC is a cooperative game, denoted as $(\mathcal{N}, \mathfrak{R})$, where $\mathcal{N}$ is the finite set of players (i.e., UEs) and $\mathfrak{R}$ is the real-valued coalition payoff function. For every coalition $\mathcal{F}_k$, $\mathfrak{R}(\mathcal{F}_k)$ is a non-negative real number from the powerset of N, which characterizes the total gain contributed by the entire coalition $\mathcal{F}_k$.*

Consider a coalition $\mathcal{F}_k$ with $1 \le k \le S$, the computation overhead of UE n is given by

$$Z_n(a_n) = \lambda_n^t \left(\frac{\beta_n}{f_n} + \frac{\alpha_n}{R_n} \right) + \lambda_n^e \frac{p_n}{\varsigma_n} \frac{\alpha_n}{R_n}; \tag{11}$$

where the offloading rate R_n in this case (i.e., $a_{nk} = 1$) can be written as

$$R_n = R_{nk} = B \log_2 \left(1 + \frac{p_{nk} h_{nk}}{\sum\limits_{j \in \mathcal{F}_k : b_k(j) < b_k(n)} p_{jk} h_{jk} + n_0} \right). \tag{12}$$

The total computation overhead induced by all the UEs in the coalition $\mathcal{F}_k$ is given as

$$Z_{\mathcal{F}_k}^r = \sum_{n \in \mathcal{F}_k} Z_{\mathcal{F}_k(n)}^r = \sum_{n \in \mathcal{F}_k} \left[\frac{\lambda_n^t \beta_n}{f_n} + \left(\lambda_n^t + \lambda_n^e \frac{p_n}{\varsigma_n} \right) \frac{\alpha_n}{B \log_2 \left(1 + \frac{p_{nk} h_{nk}}{\sum\limits_{j \in \mathcal{F}_k : b_k(j) < b_k(n)} p_{jk} h_{jk} + n_0} \right)} \right]. \tag{13}$$

Thus, the utility of coalition k that is substituted by the UEs $n \in \mathcal{F}_k$, denoted by $\mathfrak{R}(\mathcal{F}_k)$, can be defined as the following:

$$\mathfrak{R}(\mathcal{F}_k) = \sum_{n \in \mathcal{F}_k} \mathfrak{R}_n(\mathcal{F}_k) = \sum_{n \in \mathcal{F}_k} \left(Z_n^l - Z_{\mathcal{F}_k}^r(n) \right). \tag{14}$$

Here, the utility of the coalition $\mathcal{F}_k$ is the total computation offloading gain that can be obtained by utilizing the subcarrier k. It is obvious that for any coalition $\mathcal{F}_k$ with $S+1 \le k \le S+N$, we have $\mathfrak{R}(\mathcal{F}_k) = 0$, i.e., the UEs do not have any benefit with local execution and their appropriate selections of offloading decision and subcarrier allocation can further improve the system utility by reducing the computation overhead.

In the following, we formally define the coalitional game and coalition formation (i.e., coalitional structure) for computation offloading and subcarrier allocation in MC NOMA-enabled MEC networks.

Definition 2. *The coalitional game with transferable utility for computation offloading and subcarrier allocation is a triplet $(\mathcal{N}, \mathfrak{R}, \mathcal{F})$, where:*

- *$\mathcal{N}$ is the set of UEs (i.e., game players).*
- *$\mathfrak{R}(F_j)$ is the utility for every coalition $F_j \subseteq \mathcal{N}$, which includes computation gain of all the UEs in the coalition F_j.*
- *The coalitional structure is shaped as $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_{S+N}\}$, where $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for any $i \neq j$, and $\bigcup_{j=1}^{S+N} \mathcal{F}_j = \mathcal{N}$.*
- *The strategies of each player is to make a decision on computation mode (i.e., local or remote execution) and on subcarrier used for computation offloading, which is based on its computation gain as well as those of other players in the current and new coalition.*

B. Coalition Game based Algorithm

In this subsection, we develop a distributed algorithm based on the introduced game model. The most important aspect of the coalitional game setting is the formation of coalitions. Specifically, each UE has different preferences over potential coalitions and adopts the preference relation to compare any two collections of coalitions. In this regard, we present the following definition of preference relation [34].

Definition 3. *For any UE $n \in \mathcal{N}$, the preference order $\succeq_n$ is defined as a complete, reflexive, and transitive binary relation over the set of all coalitions that UE n can possibly form.*

UEs form the coalitions in order to lower the total computation overhead. Each UE in the game can decide to join/leave a specific coalition based on its preference relation and the UE's individual computation offloading gain. $\mathcal{F}_1 \succeq_n \mathcal{F}_2$ indicates that UE n prefers becoming a member of $\mathcal{F}_1$ rather than becoming a member of $\mathcal{F}_2$, where $\mathcal{F}_1 \subseteq \mathcal{N}$ and $\mathcal{F}_2 \subseteq \mathcal{N}$. Using the asymmetric counterpart of $\succeq_n$, denoted by $\succ_n$, $\mathcal{F}_1 \succ_n \mathcal{F}_2$ implies that the UE n strictly prefers becoming a member of $\mathcal{F}_1$ instead of being a member of $\mathcal{F}_2$. Since the switching process should be based on coalition-value orders (e.g., the utilitarian order) as a means to minimize the total computation overhead and each offloading UE should have a non-negative computation gain, for any UE $n \in \mathcal{N}$ and $n \in \mathcal{F}_s, \mathcal{F}_k$, we propose the following preference

$$\begin{aligned} &\mathcal{F}_s \succ_n \mathcal{F}_k \Leftrightarrow \\ &\mathfrak{R}(\mathcal{F}_s) + \mathfrak{R}(\mathcal{F}_k \setminus n) > \mathfrak{R}(\mathcal{F}_s \setminus n) + \mathfrak{R}(\mathcal{F}_k), \mathfrak{R}_j(\mathcal{F}_m) \ge 0 \\ &\mathfrak{R}_j(\mathcal{F}_m \setminus n) \ge 0, \forall j \in \{\mathcal{F}_m \setminus n\}, m = s, k. \end{aligned} \tag{15}$$

This definition implies that when the total computation gain achieved in $\mathcal{F}_s$ is greater than in $\mathcal{F}_k$ and no other UE j in $\mathcal{F}_s$ and $\mathcal{F}_k$ is negatively affected by the joining of UE n, UE n has a stronger desire to become a member of $\mathcal{F}_s$ than $\mathcal{F}_k$. To form the coalitions in accordance with the preference relation in (15), the following switch rule is defined.

Definition 4. *Given a partition* $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_{S+N}\}$ *of the set of UEs, if a UE* n *leaves its current coalition* $\mathcal{F}_k$, $k \in \{1, \ldots, S+N\}$ *and joins another coalition* $\mathcal{F}_s \in \mathcal{F}$ *with* $\mathcal{F}_s \neq \mathcal{F}_k$, *the current partition* $\mathcal{F}$ *is adjusted to form a new partition* $\mathcal{F}' = \{\mathcal{F} \setminus \{\mathcal{F}_s, \mathcal{F}_k\}\} \cup \{\mathcal{F}_k \setminus \{n\}, \mathcal{F}_s \cup \{n\}\}$.

From the switch rule, we can find the offloading decisions and subcarrier allocation from any initial coalition partition by switch operations. The switch rule allows each UE to decide on leaving its current coalition $\mathcal{F}_k$ and joining a new coalition $\mathcal{F}_s$ as long as the preference relation $\mathcal{F}_s \succ_n \mathcal{F}_k$ in (15) is satisfied. In doing so, a UE will perform a switch operation if it can strictly enhance the system utility in terms of total computation overhead and without negatively affecting individual computation gains received by the other UEs. In general, the aim of our proposed coalition formation game is to find a coalitional structure (i.e., coalition partition) in order to minimize the total computation overhead rather than individual computation gains of UEs. Meanwhile, our proposed coalitional game ensures that every UE has a benefit with computation offloading, otherwise such UEs prefer to execute their computation tasks locally.

Algorithm 1 Coalition Formation Algorithm for Computation Offloading and Subcarrier Assignment.

Create a random partition $\mathcal{F}_{ini}$ of the set of UEs $\mathcal{N}$.
Set the current partition $\mathcal{F}_{cur} = \mathcal{F}_{ini}$, $iter = 0$, and $num = 0$.
repeat
  Increase $iter$ by 1: $iter = iter + 1$.
  Select a UE $n \in \mathcal{N}$ via a predetermined permutation and find its current coalition $\mathcal{F}_k \in \mathcal{F}_{cur}$.
  Uniformly randomly choose another coalition $\mathcal{F}_s \in \mathcal{F}_{cur}, \mathcal{F}_s \neq \mathcal{F}_k$.
  if The preference relation $\mathcal{F}_s \succ_n \mathcal{F}_k$ is satisfied **then**
    UE n leaves its current coalition $\mathcal{F}_k$ and joins $\mathcal{F}_s$.
    Update the current partition according to the switch rule in Definition 4 $\mathcal{F}_{cur} \leftarrow \{\mathcal{F}_{cur} \setminus \{\mathcal{F}_s, \mathcal{F}_k\}\} \cup \{\mathcal{F}_k \setminus \{n\}, \mathcal{F}_s \cup \{n\}\}$.
    Set $num = 0$.
  else
    Set $num = num + 1$.
  end if
until $\mathcal{F}_{cur}$ converges to a Nash-stable partition $\mathcal{F}_{fin}$.
Output: Offloading decisions $a_n, \forall n \in \mathcal{N}$ and subcarrier allocation $a_{n,s}, \forall n \in \mathcal{N}_{\text{off}}, s \in \mathcal{S}$.

Based on the switch rule defined above, we design a distributed coalition formation algorithm for computation offloading and subcarrier allocation as in Alg. 1, where UEs perform switch operations until the final Nash-stable partition is achieved. The proposed algorithm can be described as follows:

- First, the algorithm is initialized by selecting any random coalition partition $\mathcal{F}_{ini}$, which is then assigned to the current partition $\mathcal{F}_{cur}$. Moreover, the numbers of iterations and consecutive unsuccessful switch operations, denoted by $iter$ and num, respectively, are set to zero.
- Next, a random UE n is selected according to a predetermined permutation. UE n randomly chooses another coalition $\mathcal{F}_s$ that is different from its current coalition $\mathcal{F}_k$. The selected UE n requests the CSI on both coalitions $\mathcal{F}_s$ and $\mathcal{F}_k$ from the MeNB. Then, UE n calculates the utilities of two coalitions as well as individual computation gains. After that, UE n can make the switching decision, i.e., to switch or not to switch.
- If the preference relation of UE n defined in (15) is satisfied, UE n informs its two corresponding coalitions about that switching and the current coalition partition is updated.
- To further improve the convergence and reduce the algorithm complexity, the concept of "consecutive unsuccessful switch operations" num is utilized [23], [35]. If a switch operation is performed, num is reset to zero, otherwise $num = num + 1$. When num equals 10 times of the number of UEs, the algorithm stops and the final Nash-stable partition is achieved.

Notably, the output of Alg. 1 is a Nash-stable coalition partition, which minimizes the total computation overhead. If task I_n is handled locally, $\mathcal{F}_{S+n} = \{n\}$ and $\mathcal{F}_{S+n} \cap \mathcal{F}_k = \emptyset, \forall k \neq (S+n)$. On the other hand, if UE n decides to offload its computation task I_n through the subcarrier s, $\mathcal{F}_{S+n} = \emptyset$ and $n \in \mathcal{F}_s$. Let us give a concrete example with $S = 4$ and $N = 7$. The coalition $\mathcal{F}_2^* = \{1, 3\}$ indicates that UEs 1 and 3 are profitable from remote execution and they utilize the subcarrier 2 for computation offloading. The coalition $\mathcal{F}_6^* = \{2\}$ implies that remote execution is not beneficial for UE 2 and the computation task I_2 is executed locally.

C. Theoretical ~~analyses~~ analysis

1) Convergence: The convergence of Alg. 1 is elaborated on the following Theorem.

Theorem 1. *Regardless of the initial coalition partition* $\mathcal{F}_{ini}$, *Alg. 1 is guaranteed to reach a final partition* $\mathcal{F}_{fin}$, *which is composed of a number of disjoint coalitions.*

Proof. As the number of UEs and the number of subcarriers are finite in our proposed algorithm and each UE can select either local computing or computation offloading, the coalitions players (i.e., UEs) can form are also finite. Specifically, there are N UEs and S subcarriers, thus Alg. 1 can form at most $S + N$ coalitions for each partition. In fact, each UE can autonomously decide its potential coalitions and perform switch operations according to the preference relation in (15) and the switch rule in Definition 4, which leads to an improvement in the system utility and results in a new-and-unvisited partition. Moreover, the number of partitions of the set $\mathcal{N}$, also known as the Bell number, is finite [34].

Since each switch operation creates a new partition and the number of partitions is finite, Alg. 1 is guaranteed to reach a final Nash-stable partition $\mathcal{F}_{fin}$. This proof indicates that our proposed algorithm is highly suitable for NOMA-enabled MEC systems with massive connectivities, where any initial offloading decisions of a huge number of UEs will finally result in a Nash-stable partition. □

2) Stability: The stability of the coalition partition $\mathcal{F}_{fin}$, achieved from the convergence of Alg. 1, is now analyzed using the concept of Nash equilibrium from the hedonic coalition-partition games. We introduce the detailed definition of Nash-stable partition in the following.

Definition 5. *A coalition partition* $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_{S+N}\}$ *is Nash-stable if* $\forall n \in \mathcal{N}, n \in \mathcal{F}_s \subset \mathcal{F}, \mathcal{F}_s \succ_n \mathcal{F}_k \cup \{n\}$ *for all* $\mathcal{F}_k \subset \mathcal{F}, \mathcal{F}_s \neq \mathcal{F}_k$.

Theorem 2. *The resulting partition* $\mathcal{F}_{fin}$ *produced by Alg. 1 is Nash-stable.*

Proof. From Definition 5, a Nash-stable partition $\mathcal{F}$ expresses that no UE prefers to leave its current coalition and join another in $\mathcal{F}$ or to deviate and act non-cooperatively. To show the stability of the resulting partition $\mathcal{F}_{fin}$, we suppose that $\mathcal{F}_{fin}$ resulting from Alg. 1 is not Nash-stable. Hence, there exists a UE $n \in \mathcal{N}$ and a coalition $\mathcal{F}_k \subset \mathcal{F}_{fin}$ such that $\mathcal{F}_k \cup \{n\} \succ_n \mathcal{F}_s$, where $\mathcal{F}_s \subset \mathcal{F}_{fin}$ is the current coalition of the UE n. In other words, UE n prefers to switch from $\mathcal{F}_s$ to $\mathcal{F}_k$, which contradicts the assumption that $\mathcal{F}_{fin}$ is the final partition. Accordingly, the partition $\mathcal{F}_{fin}$ achieved from the convergence of Alg. 1 is Nash-stable. □

3) Complexity: In general, the complexity of Alg. 1 largely depends on the number of switch operations. In each iteration, the selected UE computes the preference of its current coalition and a potential coalition, along with individual computation gains of UEs in those two coalitions. In order to check the preference relation (15) from the MeNB, the selected UE needs to obtain its own channels as well as information obtained from the UEs in the same coalition. Once the switch operation is established, the UE switches from its current coalition to the new preferred coalition. Since only one UE is selected in each iteration, there is no more than one switch operation to be performed. Denote by N_{iter} the number of iterations, then our proposed algorithm has the computational complexity of $\mathcal{O}\left(N_{iter}\right)$. It is obvious that the complexity of Alg. 1 mainly relies on the number of iterations. As will be shown in Section IV-E and Fig. 11, the proposed algorithm yields the final Nash-stable solution with significantly lower complexity than the optimal approach.

IV. Simulation Results

A. Simulation Parameters

In this section, we carry out extensive simulations to evaluate our proposed algorithm. The simulation settings are as follows. We consider a network setting with an MEC server (i.e., an MeNB), which has the coverage radius of 500 m. All the UEs are positioned at random locations within the coverage of the MeNB and the minimum distance between a UE and the MeNB is set to be 5 m. The noise power n_0 is -100 dBm, the transmit power is 100 mW, and the subcarrier bandwidth B is 1 MHz. The pathloss from a UE to the MeNB for the distance d can be computed as $L(d) = 15.3 + 37.6 \log 10(d)$. Regarding the computation model, we adopt facial recognition applications for all the UEs, which have $\alpha_n = 420$ KB and $\beta_n = 1000$ Megacycles [32], $\forall n \in \mathcal{N}$. The local computing capability f_n^l of the UE n is picked at random from the set $\{0.5, 0.8, 1.0\}$ GHz and the MEC server allocates an amount of 1.0 GHz computing resources to each offloading UE, i.e., $f_n = 10^9$, $\forall n \in \mathcal{N}$. The UE weights are set as $\lambda_n^t = 0.5$ and $\lambda_n^e = 0.5$, $\forall n \in \mathcal{N}$. The simulation result of each experiment is achieved from 500 random channel instances on average.

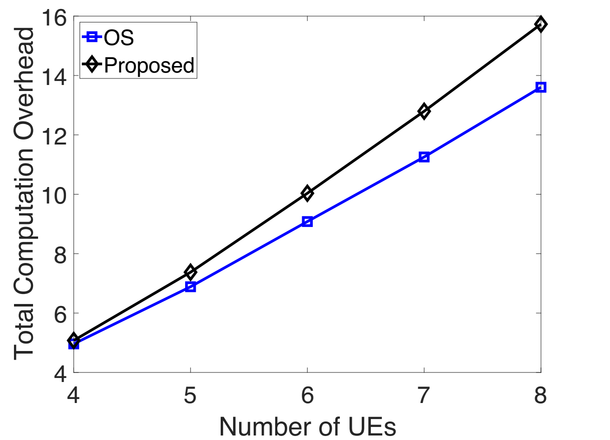

Fig. 3: Total computation overhead of the proposed approach and the optimal scheme with different numbers of UEs.

For the purpose of showing the advantages of our proposed coalition formation algorithm in reducing the total computation overhead, three following schemes are used for comparison.

- **Local Computing Only** (LCO): All the computation tasks are processed locally (local computing), i.e., $a_{ns} = 0, \forall n \in \mathcal{N}, s \in \mathcal{S}$.
- **Computation Offloading Only** (COO): all the computation tasks are executed remotely by the MEC server, i.e., $a_n = 1, \forall n \in \mathcal{N}$. This scheme can be obtained by executing our proposed algorithm with only S coalitions and the following preference relation:

$$\begin{aligned}&\mathcal{F}_s \succ_n \mathcal{F}_k \\ &\Leftrightarrow \mathfrak{R}\left(\mathcal{F}_s\right) + \mathfrak{R}\left(\mathcal{F}_k \setminus n\right) > \mathfrak{R}\left(\mathcal{F}_s \setminus n\right) + \mathfrak{R}(\mathcal{F}_k). \quad (16)\end{aligned}$$

- **Heuristic Orthogonal Offloading** (HOO): this scheme follows the orthogonal design, where each subcarrier is allocated to at most one UE. Typically, the number of subcarriers S is smaller than the number of UEs N, thus S UEs that achieve the highest individual computation gains are allowed to utilize S subcarriers while the other UEs execute their computation tasks locally.

B. Performance Comparison With the Optimal Scheme

In the first experiment, we compare the performance in terms of computation overhead between our proposed ap-

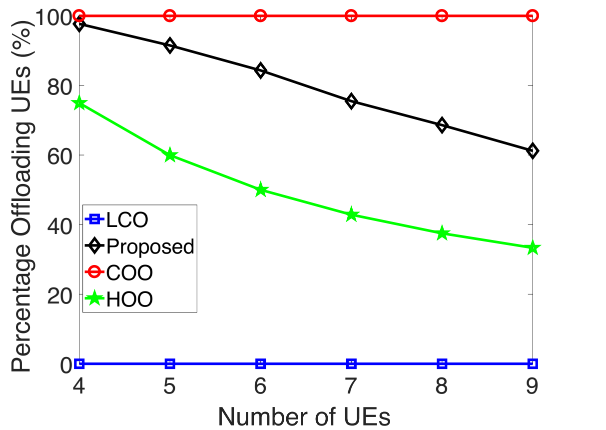

Fig. 4: Offloading percentage comparison of four schemes with different numbers of UEs.

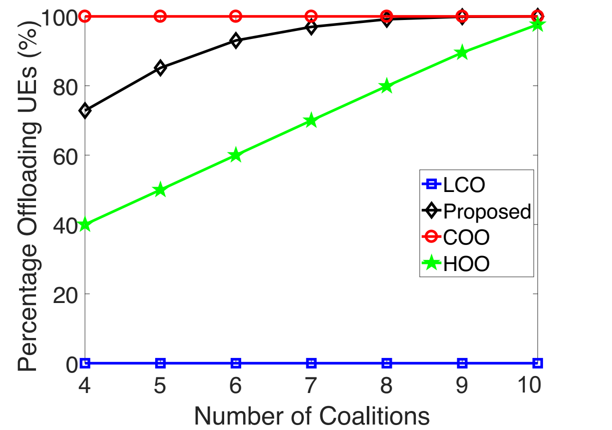

Fig. 5: Offloading percentage comparison of four schemes with different numbers of subcarriers.

proach and the optimal scheme, in which all feasible combinations of local computing and remote execution are assessed and the optimal solution is achieved by finding the smallest value of total computation overhead. It is worth noting again that each UE can select between either local computing or remote execution, whereas computation offloading can be performed through one among S possible subcarriers. For given N UEs and S subcarriers, the computational complexity associated with the exhaustive search is $\mathcal{O}\left(N^{S+1}\right)$. Due to the exponential computational complexity of exhaustive search method, it is totally impractical to achieve the optimal solution for large-scale NOMA-enabled MEC systems with a very large number of UEs. Therefore, in this simulation, we consider only 3 coalitions while varying the number of UEs between 4 and 9. Fig. 3 shows that our proposed algorithm can perform close to the optimal scheme. To further demonstrate the performance optimality of our coalition game based algorithm, we measure the average reduction in computation overhead by the optimal solution compared with our proposed algorithm as follows:

$$\text{Average Deviation} = \frac{1}{5}\sum_{N=4}^{8}\frac{Z_{\text{CG}}(N) - Z_{\text{OS}}(N)}{Z_{\text{OS}}(N)}, \qquad (17)$$

where N denotes the number of UEs and $Z_{\text{CG}}(N)$ and $Z_{\text{OS}}(N)$ denote the total computation overhead created by the proposed algorithm and the optimal solution method, respectively. Specifically, the average deviation between two algorithms is about 11.20%. Another observation is that the performance gap between our proposed algorithm and the optimal solution becomes minimal when the numbers of UEs and subcarriers are equal. We take $S = 3$ and $N = 4$ as an example, the performance gap is only $100 \times (5.077 - 4.962)/4.962 = 2.317\%$. Combined with the distributed nature, this result corroborates the advantages of our proposed game-based algorithm.

C. Percentage of Offloading UEs

To verify our proposed coalition game based algorithm, we plot the percentage of offloading UEs as a function of the number of UEs or subcarriers. Actually, the offloading percentage, denoted by p_{off}, is the ratio of the number of offloading UEs to the total number UEs, i.e., $p_{\text{off}} = \sum_{n\in\mathcal{N}} x_n/N$. Fig. 4 plots the percentage offloading of different schemes when the number of coalitions is set to be 3 and the number of UEs varies from 4 to 9. Followed the definitions of COO and LOO, their percentages of offloading UEs are 100% and 0%, respectively, regardless of the number of UEs. It indicates that the COO (LCO) scheme forces all UEs to perform computation offloading (local computing). Therefore, some UEs may be not beneficial from remote execution (in the COO scheme), while some UEs have to handle the tasks locally even though they can benefit from computation offloading (in the LCO scheme). Compared with the proposed algorithm, these two schemes are simpler and easier to implement. However, as will be shown later in Figs. 6, 7, 8, 9, and 10, their performance is typically not good as our proposed algorithm, especially for the COO (LCO) scheme in the case of massive connectivities (redundant radio resources). From the figure, we can see that for a given number of coalitions/subcarriers, the offloading percentage reduces with the number of UEs. When the number of UEs varies from 4 to 9, the offloading percentage drops from 98% to 61% for our proposed approach and from 75% to 33% for the HOO scheme. The reason is that for the NOMA-enabled MEC system with a small number of UEs, computation offloading is highly favorable, thus yielding a high offloading percentage. In contrast, UEs must compete with more other UEs for preferred subcarriers over a limited amount of radio resources. We note that in the proposed algorithm and HOO scheme, computation offloading is performed only when UEs have benefits from remote execution, otherwise UEs prefer to locally handle their tasks. Therefore, our proposed algorithm enables more UEs to benefit from remote execution than the HOO scheme, which demonstrates effectiveness of the coalition game approach and the outperformance of NOMA over OMA.

Similarly, in Fig. 5, the number of UEs is set to be 10 while the number of coalitions varies between 4 and 8. With the same number of UEs, the more coalitions we have, the more percentage of offloading UEs we get. This is because UEs can enjoy greater freedom of choice with increasing coalitions. Regarding our proposed algorithm, the offloading

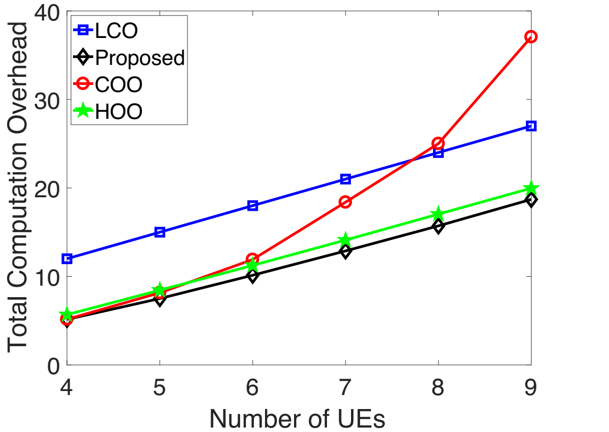

Fig. 6: Comparison of total computation overhead among four schemes under different numbers of UEs.

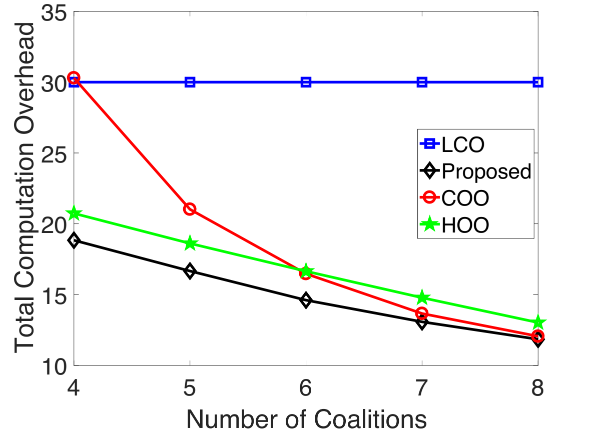

Fig. 7: Comparison of total computation overhead among four schemes under different numbers of coalitions.

percentage increases from 76% to almost 100% when the number of coalitions varies from 4 to 8. Once again, our proposed algorithm grants benefit from remote execution to more UEs than the HOO scheme.

D. *Total Computation Overhead*

Next, we evaluate our proposed algorithm in terms of total computation overhead with different numbers of UEs and subcarriers. The performance achieved by our coalition game based algorithm is also compared with three other schemes: LCO, COO, and HOO.

In Fig. 6, the number of coalitions/subcarriers is set to be 3 (i.e., $S = 3$) and the number of UEs varies from 4 to 9. As observed from Fig. 6, the proposed coalition game based approach helps reduce the total computation overhead. Particularly, our algorithm has the smallest value in terms of computation overhead in comparison with three other schemes. As the number of UEs becomes larger, the total computation overhead of all the schemes increases. In the case of the COO scheme, the total computation overhead increases dramatically when the number of UEs gets larger. It is because that each UE needs to share the same resource with more UEs on average when the number of UEs keeps increasing, thus increasing the intra-coalition interference. Therefore, UEs are more severely affected and have lower offloading possibilities. In other words, some UEs are enforced to select remote execution instead of local computing even though they do not have any benefit from remote execution, i.e., local computing makes the computation overhead smaller. Another observation is that the COO algorithm is worse than the LCO algorithm when the number of UEs gets larger.

In Fig. 7, the number of UEs is fixed at 10 and the number of coalitions varies from 4 to 8. As the figure shows, the proposed coalition game formulation outperforms all the compared schemes. Using the same calculation as in (17), averagely the proposed scheme outperforms the LCO, COO, and HOO schemes about 250%, 76%, and 53%, respectively. Obviously, the total computation overhead decreases as the number of coalitions increases. The reason is simply that UEs have higher probabilities to utilize their preferred subcarriers

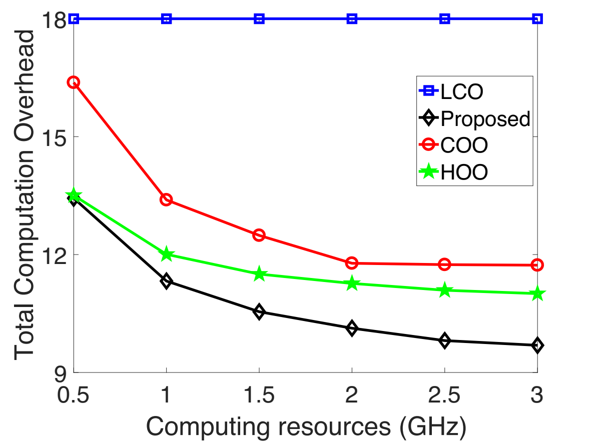

Fig. 8: Comparison of total computation overhead among four schemes under variant remote computing resources.

for computation offloading. It should be noted that the LCO always gets the constant computation overhead since all the UEs are required to handle their tasks locally, even though computation offloading is beneficial. On the other hand, when the number of coalitions becomes larger, the performance gap between the proposed algorithm and the COO and HOO schemes gets smaller. It is reasonable since each offloading UE may not need to share its preferred subcarrier with the other UEs. Accordingly, the effect of intra-coalition interference is relieved and the uplink transmission time is therefore smaller due to the higher offloading data rate.

To further obtain the performance comparison in terms of computation overhead, we set the numbers of coalitions and UEs to be 3 and 6, respectively. Fig. 8 indicates that the total computation overhead of four algorithms decreases with the amount of computing resources f_n varied from 0.5 to 3 GHz that is allocated to the offloading UEs by the MEC servers. We can see that the performance curves reduce slowly with the increment of computing resources. It is reasonable since the reduction of remote completion time would be less, which is opposed to the case of small remote computing resources, where the MEC completion time is comparable to the uplink transmission time and local completion time as well.

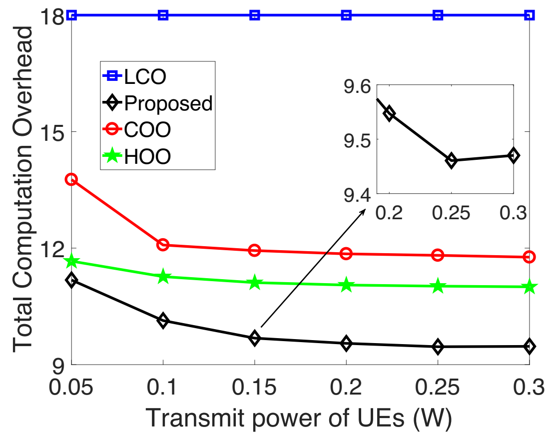

Fig. 9: Comparison of total computation overhead among four schemes under different transmit power of UEs.

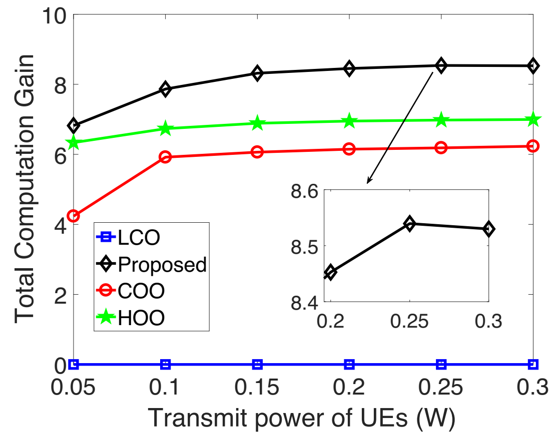

Fig. 10: Total computation gain of four schemes with different transmit power of UEs.

Compared with three other schemes, our algorithm generates the lowest computation overhead. While the COO and HOO create smaller computation overhead as the amount of remote computing resources increases, the value of the LCO scheme remains unchanged as the result that all UEs perform their computation tasks locally regardless of $f_n, \forall n \in \mathcal{N}$. When the amount of computing resources f_n is set to be 3 GHz, our proposed algorithm produces an improvement of 11.96% and 17.38% over the HOO and COO, respectively.

In Fig. 9, we plot the total computation overhead when the transmit power of UEs varies from 50 to 300 mW. From the figure, we can observe that the computation overhead decreases as the transmit power of UEs increases. It is because that the achievable transmission rate is higher with the transmit power of UEs, thus reducing the uplink transmission time and the remote computation overhead accordingly. Moreover, with the transmit power of UEs increased, the decrease of computation overhead becomes slighter. The reason is that as the transmit power of UEs increases, the increase in achievable rate is smaller. However, when the transmit power of UEs is sufficiently large, the inter-coalition interference is more severe and the transmission rate starts dropping, thus increasing the computation overhead. For example, when the transmit power p_n is set to be 200, 250, and 300 mW, the total computation overhead is 9.5474, 9.4606, and 9.4702, respectively. Comparing four schemes, the computation overhead created by our proposed coalition game based algorithm is much lower than other schemes. When the transmit power $p_n = 300$ mW, the computation overhead of our algorithm is lower than that of the HOO and COO about 13.93% and 19.53%, respectively.

Fig. 10 shows similar observations to Fig. 9. In particular, the total computation gain of the proposed algorithm, COO, and HOO increases as the transmit power of UEs increases and starts declining when the transmit power is large enough. Note that the computation gain of a UE is defined as *zero* in the case of local computing and as the *difference* between remote and local computation overhead in the case of remote execution. In addition, our proposed algorithm achieves the largest computation gain compared with three other schemes. When the transmit power of UEs is equal to 250 mW, the total computation gain of our proposed algorithm is larger than that of the HOO and COO about 22.36% and 38.05%, respectively.

E. Convergence Rate

Finally, we change the number of UEs from 4 to 9 and vary the number of coalitions between 3 and 5 to analyze the convergence of our proposed algorithm. Fig. 11 shows the average number of switch operations required for the proposed algorithm to reach the final Nash-stable partition $\mathcal{F}_{fin}$. As illustrated in the figure, the number of switch operations becomes higher with the increment of the number of UEs. In the case of 4 coalitions (i.e., 4 subcarriers), the average number of switch operations is between 4 and 8. When the number of UEs increases from 5 to 9, the number of switch operations just raises by 1. As aforementioned, the exhaustive search method needs to check N^{S+1} possible cases to obtain the optimal solution. Thus, the number of switch operations for the cases of 3, 4, and 5 coalitions is from 256 to 6561, 1024 to 59049, and 4096 to 531441, respectively. It is also depicted in Fig. 11 that with the same number of UEs, the more coalitions we have, the more switch operations the algorithm is required to perform to find the final solution. The reason for this result is that UEs have a wider range of selection alternatives for a subcarrier with better channel condition so as to reduce the uplink transmission time, hence causing an increase in the number of switch operations. From the above observations, our proposed algorithm greatly reduces the computational complexity compared with the optimal solution method and converges rapidly to the final Nash-stable partition.

V. Conclusion

In this paper, we have utilized coalitional games to develop a cooperative scheme of computation offloading in NOMA-enabled MEC systems. The optimization problem has been modeled as a coalition formation game. Based on the introduced game model, we have proposed a distributed algorithm and demonstrated that the solution achieved by our proposed algorithm is convergent and stable. Through extensive numerical results under various simulation settings, we have

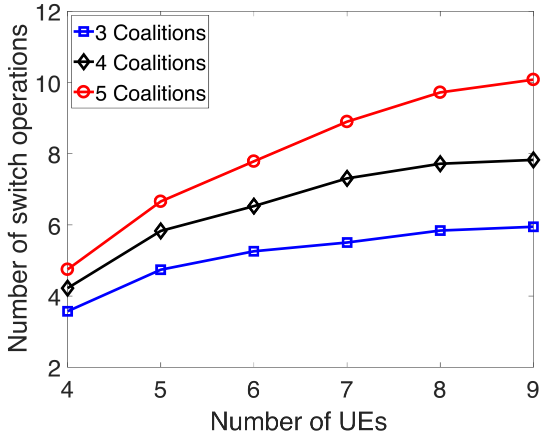

Fig. 11: Convergence of the proposed algorithm versus the number of switch operations under different numbers of UEs.

illustrated that our proposed coalition game based algorithm is competitive to the optimal scheme by exhaustive search as well as outperforming three baseline algorithms in terms of computation overhead, offloading percentage, and number of switch operations.

REFERENCES

[1] T.-H. Vo, Z. Ding, Q.-V. Pham, and W.-J. Hwang, "Access control and pilot allocation for machine-type communications in crowded massive MIMO systems," *Symmetry*, vol. 11, no. 10, Oct. 2019.

[2] "Future mobile data usage and traffic growth," https://www.ericsson.com/en/mobility-report/future-mobile-data-usage-and-traffic-growth, online; accessed 17 May 2019.

[3] ETSI. (2017) Multi-access edge computing. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

[4] Q.-V. Pham, F. Fang, H.-N. Vu, M. Le, Z. Ding, L. B. Le, and W.-J. Hwang, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *arXiv preprint arXiv:1906.08452*, 2019. [Online]. Available: http://arxiv.org/abs/1906.08452

[5] J.-W. Ryu, Q.-V. Pham, H. N. T. Luan, W.-J. Hwang, J.-D. Kim, and J.-T. Lee, "Multi-access edge computing empowered heterogeneous networks: A novel architecture and potential works," *Symmetry*, vol. 11, no. 7, Jul. 2019.

[6] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 810–819, May 2017.

[7] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Transactions on Computers*, vol. 67, no. 9, pp. 1287–1300, Sep. 2018.

[8] Q.-V. Pham and W.-J. Hwang, "α-fair resource allocation in non-orthogonal multiple access systems," *IET Communications*, vol. 12, no. 2, pp. 179–183, Jan. 2018.

[9] W. Shin, M. Vaezi, B. Lee, D. J. Love, J. Lee, and H. V. Poor, "Non-orthogonal multiple access in multi-cell networks: Theory, performance, and practical challenges," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 176–183, Oct. 2017.

[10] S. M. R. Islam, N. Avazov, O. A. Dobre, and K. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 721–742, Second Quarter 2017.

[11] F. Wang, J. Xu, and Z. Ding, "Optimized multiuser computation offloading with multi-antenna NOMA," in *IEEE Globecom Workshops (GC Wkshps)*, Singapore, Dec. 2017.

[12] A. Kiani and N. Ansari, "Edge computing aware NOMA for 5G networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1299–1306, Apr. 2018.

[13] Z. Ding, J. Xu, O. A. Dobre, and H. V. Poor, "Joint power and time allocation for NOMA–MEC offloading," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6207–6211, Jun. 2019.

[14] Z. Yang, J. Hou, and M. Shikh-Bahaei, "Energy efficient resource allocation for mobile-edge computation networks with NOMA," in *IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, Dec. 2018.

[15] Z. Yang, J. Hou, and M. Shikh-Bahaei, "Resource allocation in full-duplex mobile-edge computing systems with NOMA and energy harvesting," *CoRR*, vol. abs/1807.11846, 2018. [Online]. Available: http://arxiv.org/abs/1807.11846

[16] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay minimization for NOMA-MEC offloading," *IEEE Signal Processing Letters*, vol. 25, no. 12, pp. 1875–1879, Dec. 2018.

[17] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 375–390, Jan. 2019.

[18] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 244–12 258, Dec. 2018.

[19] Y. Wu, L. P. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 392–407, Jun. 2019.

[20] X. Diao, J. Zheng, Y. Cai, X. Dong, and X. Zhang, "Joint user clustering, resource allocation and power control for noma-based mobile edge computing," in *10th International Conference on Wireless Communications and Signal Processing (WCSP)*, Hangzhou, China, Oct. 2018.

[21] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal task assignment and power allocation for NOMA mobile-edge computing networks," *arXiv preprint arXiv:1904.12389*, 2019.

[22] Z. Yang, C. Pan, J. Hou, and M. Shikh-Bahaei, "Efficient resource allocation for mobile-edge computing networks with noma: Completion time and energy minimization," *IEEE Transactions on Communications*, 2019, in press.

[23] Y. Chen, B. Ai, Y. Niu, K. Guan, and Z. Han, "Resource allocation for device-to-device communications underlaying heterogeneous cellular networks using coalitional games," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4163–4176, Jun. 2018.

[24] M. Thi, T. Huynh, and W. Hwang, "Proportional selection of mobile relays in millimeter-wave heterogeneous networks," *IEEE Access*, vol. 6, pp. 16 081–16 091, Mar. 2018.

[25] W. Saad, Z. Han, T. Basar, A. Hjorungnes, and J. B. Song, "Hedonic coalition formation games for secondary base station cooperation in cognitive radio networks," in *IEEE Wireless Communication and Networking Conference*, Sydney, Australia, Apr. 2010.

[26] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7244–7257, Nov. 2016.

[27] M. S. Ali, H. Tabassum, and E. Hossain, "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems," *IEEE Access*, vol. 4, pp. 6325–6343, Aug. 2016.

[28] Q.-V. Pham, T. L. Anh, N. H. Tran, B. J. Park, and C. S. Hong, "Decentralized computation offoading and resource allocation for mobile-edge computing: A matching game approach," *IEEE Access*, vol. 6, pp. 75 868–75 885, Nov. 2018.

[29] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[30] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[31] F. Wang, "Computation rate maximization for wireless powered mobile edge computing," in *23rd Asia-Pacific Conference on Communications (APCC)*, Perth, Australia, Dec. 2017.

[32] Q.-V. Pham, L. B. Le, S.-H. Chung, and W.-J. Hwang, "Mobile edge computing with wireless backhaul: Joint task offloading and resource allocation," *IEEE Access*, vol. 7, pp. 16 444–16 459, Jan. 2019.

[33] Y. Pochet and L. A. Wolsey, *Production planning by mixed integer programming*. B-1348 Louvain-la-Neuve, Belgium: Springer Science & Business Media, 2006.

[34] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game theory in wireless and communication networks: theory, models, and*

applications. The Edinburgh Building, Cambridge CB2 8RU, UK: Cambridge University Press, 2012.

[35] Y. Chen, Y. Niu, B. Ai, Z. Zhong, D. Wu, and K. Li, "Using coalition games for QoS aware scheduling in mmwave WPANs," in *IEEE 87th Vehicular Technology Conference (VTC Spring)*, Porto, Portugal, Jun. 2018.

Quoc-Viet Pham (M'18) received the B.S. degree in electronics and telecommunications engineering from Hanoi University of Science and Technology, Vietnam, in 2013, and the M.S. and Ph.D. degrees, both in telecommunications engineering, from Inje University, South Korea, in 2015 and 2017 respectively. He is currently a research professor at High Safety Core Technology Research Center, Inje University, South Korea. Prior to this position, he was a research fellow at ICT Convergence Center, Changwon National University, South Korea. He received the best PhD dissertation award in Engineering from Inje University in 2017. His research interests include convex optimization, game theory, and machine learning to analyze, design, and optimize resource allocation in 5G wireless and communication networks.

Hoang T. Nguyen received the B.S. degree in Electronics and Telecommunications Engineering from Posts and Telecommunications Institute of Technology, Vietnam, in 2017, and the M.S degrees in Information and Communications Systems from Inje University, South Korea, in 2019. His research interests include game theory and optimization in 5G wireless and communication networks.

Zhu Han (S'01-M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. He is also a Chair professor in National Chiao Tung University, ROC. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Currently, Dr. Han is an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019 and ACM distinguished Member since 2019. Dr. Han is 1% highly cited researcher since 2017 according to Web of Science.

Won-Joo Hwang (S'01-M'03-SM'17) received the B.S. and M.S. degrees in computer engineering from Pusan National University, Pusan, South Korea, in 1998 and 2000, respectively. He received the Ph.D. degree in information systems engineering from Osaka University, Osaka, Japan, in 2002. He is currently a Full Professor at Inje University, Gimhae, South Korea. His research interests include network optimization and cross layer design. He is a senior member of the IEEE.