Uplink Scheduling in a NOMA-enabled Single-Cell Wireless Network Using Simulated Annealing

Botao Yang, Ye Liu, Chung Shue Chen

Uplink Scheduling in a NOMA-enabled Single-Cell Wireless Network Using Simulated Annealing

Botao Yang*, Ye Liu*, Chung Shue Chen†
*Dept. of Communications and Networking, Xi'an Jiaotong-Liverpool University, China
†Nokia Bell Labs, Paris-Saclay Center, 91300 Massy, France
Email: botao.yang19@student.xjtlu.edu.cn, ye.liu02@xjtlu.edu.cn, chung_shue.chen@nokia-bell-labs.com

***Abstract*—By allowing multiple users to transmit using the same frequency band at the same time, non-orthogonal multiple access (NOMA) can support more users as compared to orthogonal multiple access (OMA) given a fixed amount of time-frequency resources. In this paper, we study the resource allocation problem in the uplink of a single-cell network when NOMA is enabled, where the maximum completion time of serving all connected users is to be minimized. While such an objective function minimizes the required time to serve uplink users, the resource allocation problem is NP-hard. We propose a serial collaborative optimization framework based on simulated annealing (SA) to search for the optimal user pairing and scheduling solution. Simulation studies show that the proposed algorithm for NOMA scheduling can reduce the maximum completion time by more than 30% when compared against OMA scheduling and random NOMA user clustering.**

***Keywords*—Non-orthogonal multiple access (NOMA), resource scheduling, maximum completion time, serial collaborative optimization, simulated annealing (SA).**

I. Introduction

In 4G/5G wireless cellular systems, orthogonal frequency-division multiple access (OFDMA) is used as the major radio access technology. OFDMA is an orthogonal multiple access (OMA) scheme that avoids intra-cell interference. Despite its ease of implementation, OMA suffers from low spectral efficiency when some bandwidth resources are allocated to users with poor channel conditions [1]. With the rapid growth of demand for mobile data network capacity and transmission speed, OMA based cellular systems are unable to keep up with the applications in the next-generation radio access technology [2], , and non-orthogonal multiple access (NOMA) is considered to be a promising technique to overcome the limitations of OMA and improve system performance [3]. In contrast to OMA schemes, NOMA allows multiple users to be superimposed on the same time-frequency resource block, where successive interference cancellation (SIC) can be used to separate signals from different users. As a result, the bandwidth allocated to the users with poor channel conditions can still be used by the users with strong channel conditions, which can significantly improve the system's spectral efficiency [4].

With higher sum capacity and enhanced spectrum efficiency, NOMA provides support for more users' communication needs at the same time, thus having the potential to improve quality-of-service (QoS) for a variety of communication applications. For delay insensitive services such as email, rate constraints are relatively relaxed for individual users and throughput (i.e., the sum of data rates delivered to all terminals in a network) is the most broadly used criterion to evaluate system performance. Over the past few years, this topic has been already thoroughly explored by academia and industry [5]–[8]. On the other hand, the explosive growth of the Internet in recent years has also created many applications that are sensitive to packet delays from end-to-end. In these cases, each user must transmit a certain amount of information within a certain period and maximizing the throughput is no longer an appropriate strategy [9]. Given a number of requests from multiple devices to transmit data packets of varying lengths, efficient scheduling scheme is required to avoid long queuing at the buffers of congested links.

In terms of delay optimization, the maximum completion time (i.e., the makespan, denoted by $C_{\max}$) is a widely adopted metric. Recently, completion time reduction in NOMA-enabled systems has received attention in the research community. Specifically, the uplink NOMA scheduling problem of minimizing $C_{\max}$ is proven to be NP-hard [10], leading to high computational complexity in conventional iterative methods. To achieve completion time minimization, [11] proposed a learning-based hybrid algorithm to jointly optimize time slot and transmit power allocation. However, this work assumes that there is only one frequency channel available, which does not reflect most practical communication scenarios.

To solve this intractable problem within affordable time and complexity, heuristic algorithms with performance guarantees are preferred. Simulated annealing (SA) [12] is a metaheuristic approach which seeks for the global optimum of a given optimization problem. Compared to conventional iterative methods, the advantages of SA include its easy implementation and possibility of reaching the global optimal without being trapped in a local optimum [13]. SA scheme can provide satisfactory result with a relatively low number of iterations [14], [15], which makes it suitable for hard optimization problem.

Aiming at minimizing the maximum completion time of serving uplink users in a single-cell NOMA-enabled network, we formulate the scheduling problem as a combinatorial optimization problem with a discrete strategy set. The correlation between user clustering (or, equivalently, user grouping) and frequency channel allocation is taken by a serial collaborative optimization. In the proposed scheme, the optimizations of

This work has been supported in part by Xi'an Jiaotong-Liverpool University Research Development Fund RDF-21-01-045.

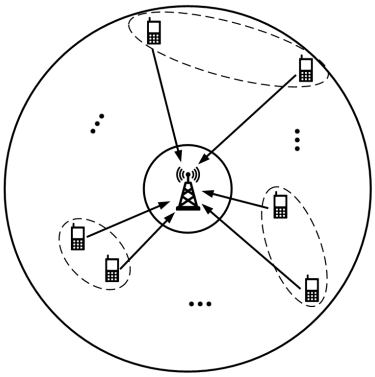

Fig. 1. User grouping in an uplink NOMA-based system, where the base station is located at the middle point.

user clustering and channel assignment scheduling are carried out in a separate manner, which significantly decreases the computational complexity. Both steps are optimized by SA but with independent parameter conditions. Numerical results show that our proposed scheme reduces more than 30% of the maximum completion time against heuristic OMA on average in the considered simulation settings.

The rest of the paper is organized as follows. In Section II, we describe the system model and problem formulation. In Section III, we present the proposed solution and optimization algorithm. In Section IV, we show the simulation result and comparison. Finally, we conclude the paper in Section V.

II. System Model and Problem Formulation

Consider the uplink of a single-cell NOMA network, where K users request to transmit data packets of varying lengths to the base station (BS), as shown in Fig. 1. The frequency resource is equally divided into F frequency channels, where each channel has a bandwidth of W Hz. The BS is equipped with a single antenna and all the K users are single-antenna devices. Each user has a maximum available transmission power of $P_{\max}$. Also, each user k transmits a single data packet of L_k bits in length with power p_k.

The wireless links experience independent and identically distributed block Rayleigh fading and additive white Gaussian noise (AWGN), where the power spectral density of the AWGN is denoted as N_0. We consider power-domain NOMA, where SIC at the receiver is used to decode signals from users that are sent in the same time-frequency block. To minimize the complexity of SIC, we assume that each frequency channel can be simultaneously occupied by at most two users. Hence, at most $K/2$ NOMA clusters can be formed, where each NOMA cluster contains two users that concurrently transmit to the serving BS via the same frequency channel. Without loss of generality, K is assumed an even number, for otherwise we can add one dummy user with 0 bit to transmit.

The channel gain between user k and the BS via channel $f \in \mathcal{F}$ is denoted by h_k^f, where $\mathcal{F}$ is the set of frequency channels and the size of $\mathcal{F}$, i.e., $|\mathcal{F}|$, is equal to F. Specifically, $h_k^f \triangleq g_k^f/\sqrt{d_k^\alpha}$, where $g_k^f \sim CN(0,1)$ is the Rayleigh fading coefficient, d_k is the distance between user k and the BS, and α is the path loss exponent. For simplicity, we assume that for any specific user k, the channel coefficient remains the same regardless of the frequency channel selected [10], i.e.[1],

$$|h_k^1|^2 = |h_k^2|^2 = \cdots = |h_k^F|^2 = |h_k|^2. \tag{1}$$

Users are sorted and indexed in ascending order according to the channel conditions, i.e.,

$$|h_1|^2 \le |h_2|^2 \le \cdots \le |h_K|^2. \tag{2}$$

It is assumed that that perfect channel state information (CSI) is available at BS and the channel coefficients are quasi-static.

When two users i and j transmit in NOMA mode, we form a NOMA cluster (i,j) and perform scheduling based on this NOMA cluster. The BS receives a linear superposition of signals from these two users, i.e.,

$$y_f = h_i\sqrt{p_i}x_i + h_j\sqrt{p_j}x_j + w, \tag{3}$$

where y_f is the received signal on the f-th frequency channel, x_k is the normalized unit-power signal transmitted by user k such that $\mathbb{E}[x_k^2] = 1$, and w denotes the noise received per channel with power N_0W. Suppose that user i is the strong user with a higher SNR, i.e., $p_i|h_i|^2 > p_j|h_j|^2$. According to the principle of SIC, the strong user i is decoded first, treating user j's signal as interference. After the strong user's signal is correctly decoded, the BS can regenerate the strong user's signal and remove that from the received composite signal. Then, the BS can decode the weak user's signal without co-channel interference from the other user. For the first decoded user (i.e., strong user i), the achievable data rate is given by

$$\begin{aligned} R_i^{\text{SIC},j} &= W\log_2\left(1+\frac{p_i|h_i|^2}{p_j|h_j|^2+N_0W}\right) \\ &= W\log_2\left(1+\frac{p_i\gamma_i}{p_j\gamma_j+1}\right), \end{aligned} \tag{4}$$

where $\gamma_i \triangleq |h_i|^2/(N_0W)$. Next, the second decoded user (weak user j) can enjoy the interference-free transmission so the achievable rate of user j equals to R_j^{OMA}, i.e.,

$$R_j^{\text{OMA}} = W\log_2\left(\frac{1+p_j|h_j|^2}{N_0W}\right) = W\log_2(1+p_j\gamma_j). \tag{5}$$

With the above, the required transmission time of user j on any frequency channel f in OMA is given by

$$\tau_j = \frac{L_j}{R_j^{\text{OMA}}}. \tag{6}$$

Two users within a NOMA cluster start transmitting at the same time. As illustrated in Fig. 2, the time required to finish the transmission of the two-user NOMA cluster (i,j) depends on which user finishes sending its packet first, i.e.,

$$u_{i,j} = \begin{cases} \tau_j, & \text{if } \frac{L_i}{R_i^{\text{SIC},j}} < \tau_j \quad \text{(case 1)}, \\ \tau_j + \frac{L_i - R_i^{\text{SIC},j}\tau_j}{R_i^{\text{OMA}}}, & \text{otherwise} \quad \text{(case 2)}. \end{cases} \tag{7}$$

Note that in the above case 1, the required transmission time of the NOMA cluster is the same as that of the weak user j,

[1]Fixing the NOMA user clustering, the remaining scheduling problem is still NP-hard even with the assumption in (1); see Section II-B.

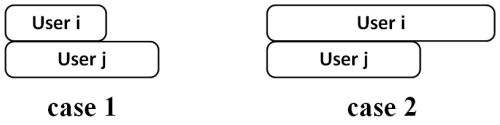

Fig. 2. Illustration of the processing time of two users in a NOMA cluster.

since user j may have a long packet to send and/or the OMA data rate of user j is low. In case 2, the transmission rate of the strong user i has the opportunity to be from $R_i^{\text{SIC},j}$ to R_i^{OMA} when the weak user j finishes transmission. While some signaling may be needed to realize such a data-rate increase, here we use the model in case 2 of (7) to study the potential gain using NOMA in ideal situations, ignoring any potential overhead that may be necessary in practice.

It is evident from (4) and (5) that within a NOMA cluster (i, j), by letting the strong user i transmit at a power level $p_i = P_{\max}$, the processing time of user i (τ_i in OMA or, when using NOMA, $\frac{L_i}{R_i^{\text{SIC},j}}$ in case 1 or $u_{i,j}$ in case 2) will be minimized. While p_j affects the processing time of users i and j and would be subject to optimization, the value of p_j and the user clustering decision are intertwined. Specifically, the user clustering decision needs to take the users' power and channel gain profiles into account, such that finding a suitable weak user to form a cluster with a strong user is similar to adjusting the transmission power of the weak user in an existing NOMA cluster. Thus, we let $p_k = P_{\max}, \forall k$, in the rest of the paper.

In the following, we introduce the two essential elements of the makespan-minimization problem with NOMA.

A. Cluster Establishment

Denote $l \in \mathcal{L}$ as a particular clustering scheme that groups the K users into $\theta(l)$ clusters, where $\mathcal{L}$ is the set of all feasible clustering methods and each cluster may contain either one or two users, i.e., $K/2 \leq \theta(l) \leq K$. For any pair of users i and j, one observes that

$$\tau_i + \tau_j \geq u_{i,j}, \tag{8}$$

where the proof of (8) is detailed in Appendix A. Equation (8) suggests that if users i and j are transmitting on the same frequency channel, then using NOMA to serve the two users will never prolong the completion time on that channel as compared to OMA. While it is possible to reduce the maximum completion time of the network if $u_{i,j}$ takes the form of case 2 and users i and j are scheduled on different channels[2], allowing a cluster to have a single user enlarges $|\mathcal{L}|$ and may increase the complexity of solving the NOMA scheduling problem. Encouraged by the benefit of NOMA as shown in (8), we assume that when using NOMA, each cluster contains two users and $K/2$ clusters are formed.

Given a user grouping scheme, the processing time of every cluster can be determined by (7). We can index the clusters by $m \in \{1,\ 2,\ \cdots,\ K/2\}$ and sort them according to the descending order of the time required to finish the transmission for each cluster, denoted by u_m, such that

$$u_1 \geq u_2 \geq \cdots \geq u_{K/2}. \tag{9}$$

B. Spectrum and Time Resource Allocation

After the user clustering is determined, the remaining scheduling problem belongs to the class of parallel machine scheduling that is NP-hard [16], where each NOMA cluster requires a single transmission on one of the F identical but independent frequency channels.

Denote $s \in \mathcal{S}$ as a particular scheduling scheme that allocates the two-user clusters to the F frequency channels, where $\mathcal{S}$ is the set of all feasible scheduling schemes. A scheduling scheme s can be represented by a size-$\frac{K}{2}$ tuple whose elements indicate the destination channel for each cluster. Specifically, the m-th cluster is labeled to transmit using the $s(m)$-th channel, i.e.,

$$1 \leq s(m) \leq F, \quad m \in \{1,\ 2,\ \cdots,\ K/2\}. \tag{10}$$

We impose the following constraints when performing the scheduling:

1) A channel, when scheduled to carry uplink data signals from users, will keep busy until all scheduled transmissions are completed. In other words, we ignore the impact of any packet header or signaling that may exist in a particular communication protocol.
2) No cluster is allowed to occupy more than one frequency channel at the same time.
3) Once the transmission for a user starts in a particular frequency channel, the scheduler is not allowed to interrupt it until its completion.

While removing the second and the third constraints may provide further space for minimizing the maximum completion time, a more complicated communication protocol and scheduling algorithm would be needed if those constraints are not imposed. We leave the study of uplink NOMA scheduling without the above constraints to a future work.

Given the above settings, the time needed to complete the transmission at channel f is given by the sum of the completion times of the clusters that are allocated to the same channel, i.e.,

$$C_f = \sum_{m \in \mathcal{M}_f} u_m, \tag{11}$$

where $\mathcal{M}_f \triangleq \{m \,|\, s(m) = f\}$ is the set of clusters that are allocated to channel f.

Assume that the transmissions at all the channels start at time zero. The maximum completion time of the NOMA system is defined as the time required to finish transmission in all the channels, as depicted by Fig. 3, i.e.,

$$C_{\max} = \max_{f \in \mathcal{F}} \{C_f\}. \tag{12}$$

Our goal is to find the optimal user clustering and channel schedules that minimize the maximum completion time of the

[2]This can happen if $u_{i,j} > \tau_i$, $u_{i,j} > \tau_j$, and when scheduling users i and j using OMA, both user i and user j can finish earlier than the case when scheduling the NOMA cluster (i, j).

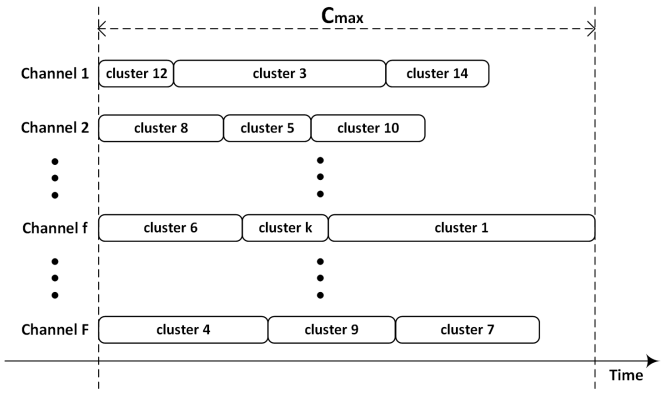

Fig. 3. Multi-channel scheduling for NOMA clusters.

NOMA system, i.e.,

$$(l^*, s^*) = \arg\min_{l\in\mathcal{L}, s\in S}\{C_{\max}(l, s)\}. \tag{13}$$

III. Optimization and Solution

A. Complexity Analysis

The problem in (13) is of combinatorial nature with a finite discrete strategy set. The total number of user clustering schemes is given by

$$|\mathcal{L}| = \frac{K!}{2^{K/2}(K/2)!} = \frac{\Pi_{i=1}^{K/2}(K/2+i)}{2^{K/2}} > (K/4)^{K/2}, \tag{14}$$

where the inequality is true because in $\frac{\Pi_{i=1}^{K/2}(K/2+i)}{2^{K/2}}$, there are $K/2$ terms in the numerator that are no smaller than $K/2$. As for the total number of channel scheduling schemes $|\mathcal{S}|$, note that the transmission order of the clusters in a channel does not affect the completion time. Therefore, $|\mathcal{S}|$ can be calculated by the Stirling number of the second kind, i.e., $S(K/2, F)$ [17]. A lower bound on $S(K/2, F)$ can be obtained as

$$S(K/2, F) > F^{K/2-F}, \tag{15}$$

where Appendix B details the proof. As a result, the size of the strategy set of our problem is lower bounded as

$$|\mathcal{L}\times\mathcal{S}| > (K/4)^{K/2}F^{K/2-F}. \tag{16}$$

Equation (16) suggests that the size of the strategy set for NOMA optimization scales up exponentially as the number of channels and serving users in the system increase. Therefore, exhaustive search for optimal solution is computationally expensive even for moderate K and F.

One class of approximation algorithms is stochastic search with iterative improvement [18]. This local search method works by continuously exploring the neighborhood of a solution in order to find a better solution. When there is a gain, the current solution is replaced by the new one. However, the NP-hardness nature of a problem can limit the performance of conventional stochastic search because such algorithms can be trapped by local minima. Besides, purely random search is known to be slow for convergence and may not be practical for the application in our context.

B. Simulated Annealing

1) Preliminary: The core mechanism of simulated annealing (SA) is based on the analogy of ideal crystal formation in thermodynamics. In the controlled annealing process, the disordered configuration of particles in high energy system gradually rearranges into regular crystalline solid state with low energy [6]. SA algorithm mimics this metallurgical method to minimize the objective function by a series of guided probabilistic random search in the strategy set.

In general, a SA follows the procedures below:

1) Select a starting point x_0 and initialize from a high temperature setting.
2) Perturb the current placement to a new state. If the change is accepted, the current state is updated to the new one. This process is repeated in an iterative manner, until thermodynamic equilibrium is reached with current temperature.
3) The temperature is then lowered, before carrying out a new series of transformation (state transitions).
4) The above loop continues until a freezing point or preset termination condition is reached.

Note that in an iteration, SA may not always select the next state that corresponds to a better solution as compared to the current state. In particular, the next state that exhibits as a worse solution than the current state can be chosen probabilistically [12]. In the minimization problem, a next state that decreases the objective function will be accepted; on the other hand, a next state that increases the objective function also has a chance of being accepted with a probability determined by a given acceptance function.

Note that the variability of the candidate solution and randomness of new state acceptance decrease as the parameter called the temperature T drops (cooling). Specifically,

1) When $T \to \infty$ (high-energy system), it behaves like a purely random walk that is always moving to a successor chosen uniformly at random from the entire strategy set.
2) When $T \to 0$ (low-energy system), it degenerates into a deterministic iterative improvement scheme that refuses any moves towards worse objective values. Note that it can get stuck on a local minimum.
3) At an intermediate temperature, the SA algorithm intermittently accepts the transformations that increase the objective function. This provides for the system to be pulled out from a local minimum.

2) Sampling Method: Sampling method is the core of SA algorithm, which governs the behavior of iteratively updating the current state depending on the evaluation of candidate states. There are two mainstream sampling methods for the trial point selection and acceptance test [19].

1) Gibbs Sampler Algorithm (GS): Denote $\mathcal{N}$ as the state set. At a temperature T, the sampler will evaluate all the possible candidate state $x \in \mathcal{N}$ and then selects a new state according to the following probability distribution:

$$p(x) = \frac{e^{-U(x)/T}}{\sum_{x'\in\mathcal{N}} e^{-U(x')/T}} \tag{17}$$

where $U(\cdot)$ is the evaluation or cost function.

2) Metropolis-Hasting Algorithm (MH): This sampling algorithm randomly selects a candidate state x' from $\mathcal{N}$ and evaluates the change to the cost function:

$$\Delta U = U(x') - U(x). \tag{18}$$

If the state transition leads to a decrease in the cost, it is unconditionally accepted. If the transition would cause an increase to the cost function, the new state will be accepted with the following probability, i.e.,

$$p_{\text{accept}}(x') = e^{-\Delta U/T}. \tag{19}$$

When $|\mathcal{N}|$ is large, the computational complexity of GS for each state update is far above the MH algorithm. Therefore, we use MH for our problem. We conduct multiple iterations at each temperature stage in MH in order to compensate its possible deficiency due to the locally incomplete sampling performed per iteration.

3) Cooling Schedule: Cooling scheme would affect the convergence speed and the quality of the SA optimization. Empirically, the temperature should decrease gradually by stages to enforce regularities of the system. It can be proven that SA converges asymptotically to the optimal solution. However, such convergence requires exponential time [13]. Hence, in practice, faster cooling schedules are used, with which SA does not guarantee to converge exactly to the global optimum and thus provides an approximate solution. According to the SA sampling mechanism, randomness of state selection and the tolerance for bad move would be decreased with temperature drop. Therefore, an appropriate cooling scheme should be able to initially search the space for global minima and later, as the system cools down, converge on a solution with proper accuracy [20]. Annealing too fast is likely to result in defects. On the other hand, annealing too slow will lead to a large number of iterations taken for convergence. Hence, to strike a balance between the convergence speed and optimality, we adopt the following geometric cooling scheme in our work:

$$T_{n+1} = \beta T_n, \quad T_n = T_0 \beta^n, \quad 0 < \beta < 1, \tag{20}$$

where T_0 is the initial temperature, β is a tunable parameter which controls the temperature drop rate, and n is the index for the temperature cooling schedule.

The SA algorithm for our problem is shown in Algorithm 1. Specifically, with a fixed temperature value, a number of I iterations are conducted. The number of new state acceptance within each I iterations is recorded by η, and the frequency of new state acceptance, calculated as $\eta \cdot I^{-1}$, is compared against a threshold value "minMove". If $\eta \cdot I^{-1} <$ minMove, it is considered that the steady state is reached (also known as the thermodynamic balance), and the SA algorithm terminates with the output x^*. Otherwise, the temperature is reduced according to (20), and another I iterations will be performed using the new temperature. The SA algorithm will also terminate if a preset maximum number of iterations is reached.

Algorithm 1 SA Optimization Algorithm (MH)

Initialization:
$x \leftarrow x_0$ starting state
$T \leftarrow T_0$ initial temperature
$x^* \leftarrow x$ the best solution found so far
repeat
  for $i = 1$ to I **do**
    $\eta \leftarrow 0$ reset counter for new state acceptance
    Randomly generate x' in the neighborhood of x
    if $U(x') < U(x)$ **then** accept it
    else
      accept it with probability $\exp\left(-\frac{U(x')-U(x)}{T}\right)$
    end if
    if x' is accepted **then**
      state transition $x \leftarrow x'$
      increase the counter $\eta \leftarrow \eta + 1$
      if $U(x) < U(x^*)$ **then**
        update the best solution ever found $x^* \leftarrow x$
      end if
    end if
  end for
until $\eta \cdot I^{-1} <$ minMove
Output: x^*

C. Serial Collaborative Optimization

In our problem, every feasible solution consists of a user grouping scheme combined with a channel scheduling scheme, which are correlated with each other, i.e.,

$$x = (l, s). \tag{21}$$

Instead of treating user clustering and resource allocation as a joint union [21], our proposed serial collaborative method decomposes the problem into two separate parts; see Fig. 4. Such a decomposition can significantly reduce the search space in each SA loop and, therefore, reduce the overall computational complexity. Note that both SA loops are implemented according to Algorithm 1 with different parameters.

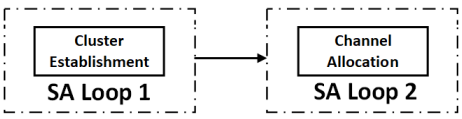

Fig. 4. Structure of SA-based serial collaborative optimization.

1) SA Loop 1 for Cluster Establishment: In this step, we perform SA optimization for user grouping scheme to minimize the sum of the processing time of all clusters. A candidate solution can be generated by the following procedure:

1) Randomly select two clusters and split them to four users.
2) Reassemble them into two different clusters. Define $\mathcal{N}_l$ as the neighborhood set of any clustering scheme l. The size of $\mathcal{N}_l$ can be found as

$$|\mathcal{N}_l| = 2 \times \binom{K/2}{2} = \frac{K(K-2)}{4}, \tag{22}$$

where $\binom{K/2}{2} = \frac{(K/2)(K/2-1)}{2}$.

The utility function is selected as the sum of the processing time of all clusters, i.e., $\sum_{m=1}^{0.5K} u_m$. While such a utility function is not equal to the objective function in (13), the utility function minimizes a measure of the processing time of all the clusters which will then impact the maximum completion time for any given schedule.

2) SA Loop 2 for Channel Allocation: With the user clustering obtained in SA loop 1, the processing time of every cluster is fixed. The remaining problem is how the tasks are distributed over the available frequency channels. In this step, we perform SA optimization for channel scheduling scheme to minimize the makespan (i.e., maximum completion time).

A candidate solution can be generated by randomly selecting two clusters in different frequency channels and swapping their position. The number of candidate solutions is maximal when clusters are equally distributed among the available frequency channels. Define $\mathcal{N}_s$ as the set of neighborhood of a scheduling scheme s. Therefore, the size of $\mathcal{N}_s$ has the following upper bound, i.e.,

$$|\mathcal{N}_s| \leq \binom{F}{2} \times \left\lceil \frac{K}{2F} \right\rceil^2 \leq \binom{F}{2} \left(\frac{K}{2F} + 1 \right)^2 = \frac{1}{8} (K + 2F)^2 \left(1 - \frac{1}{F}\right). \quad (23)$$

The utility function for SA loop 2 is selected as the maximum completion time, i.e., $\max_{f \in \mathcal{F}} C_f$.

3) Initialization: In both SA loops, the state generation schemes allow that every single state can be reached from any other state in a number of moves. Therefore, the initial solution can be selected randomly to start the iterative search.

IV. Simulation and Numerical Results

A. Simulation Experiment Design

To test performance, we produce the simulation results by randomly generating 1000 network topologies and then averaging the makespan from all the topologies. Specifically, in each topology, each user is randomly scattered within a bounded annular region centered at the BS, where the user-BS distance d_k is between 100 meters and 1000 meters. The maximum allowed user transmission power is set to be $P_{\max} = 23$ dBm. The path loss exponent is set to $\alpha = 3$. The packet size of each user is randomly chosen in ranges that are specified in Table I. Parameters related to the SA algorithm can be found in Table II.

In order to provide some benchmark results in comparison with the serial collaborative algorithm, the maximum completion time is also evaluated for random NOMA clustering (i.e., randomly select two users for each cluster) as well as conventional OMA scheme scheduled by the longest-processing-time-first (LPT) rule. The LPT rule is a heuristic method for the NP-hard identical machine scheduling problem [16]. The LPT rule works as follows:

1) Order the jobs (i.e., the users in this paper's context) by descending order of their processing time.
2) Schedule each job in this sequence into a machine (i.e., a frequency channel) in which the current flowtime (total processing time of the scheduled jobs) is the shortest.

Table I: Parameter settings for network topology generation

User-to-BS distance	$100\text{m} \leq d_k \leq 1000\text{m}$
Max device transmit power	$P_{\max} = 23$ dBm $(0.2W)$
Noise spectral density	$N_0 = -174$ dB/Hz
Path loss exponent	$\alpha = 3$
Bandwidth per channel	$W = 180$ kHz
Data packet size (in bits)	$[10^4, 10^8]$, $[10^4, 10^5]$, or $[10^4, 4 \times 10^4]$

Table II: Parameter settings for SA

SA Parameters	SA Loop 1	SA Loop 2
T_0	150	100
I	200	200
minMove	5%	5%
β	0.95	0.95

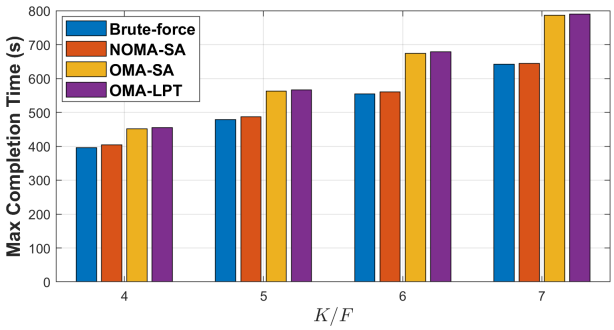

Fig. 5. Makespan versus K/F ratio ($F = 2$).

The LPT heuristic tries to place the shorter jobs towards the end of a schedule, where they can be used for balancing the loads. In the worst case, the makespan of LPT schedule is no longer than $4/3$ times that of the optimal schedule [16], i.e.,

$$\frac{C_{\max}(\text{LPT})}{C^*_{\max}} \leq \frac{4}{3} - \frac{1}{3F}, \quad (24)$$

where $C_{\max}(\text{LPT})$ is the makespan obtained by LPT.

B. Numerical Results

We firstly compare the results of the proposed algorithm against brute-force search in small networks, where $F = 2$ and $K \in \{4F, 5F, 6F, 7F\}$. In these scenarios, the NOMA-SA algorithm always finds the optimal clustering scheme that minimizes the sum of processing time that is evaluated in SA loop 1. When comparing the makespan, Fig. 5 indicates that the NOMA-SA algorithm succeeds in finding solutions that perform very closely to global optima, and OMA-SA can increase the makespan by 22% as compared to NOMA-SA.

The makespan versus K for $F = 4,\ 8,$ and 12 are depicted in Fig. 6, Fig. 7, and Fig. 8, respectively, where the packet size of each user varies randomly from 10^4 bits to 10^8 bits. Note that "RAND" in the figures refers to random user clustering, such that the SA loop 1 optimization is not performed but

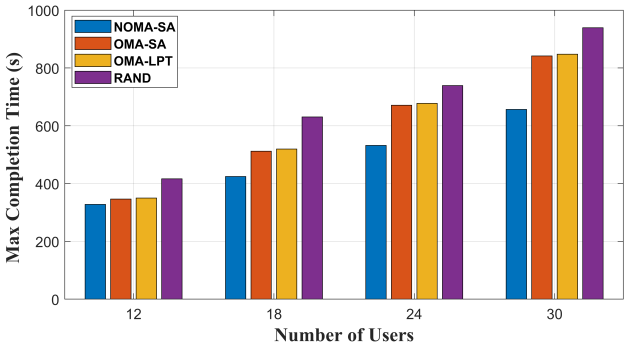

Fig. 6. Makespan versus the number of users ($F = 4$).

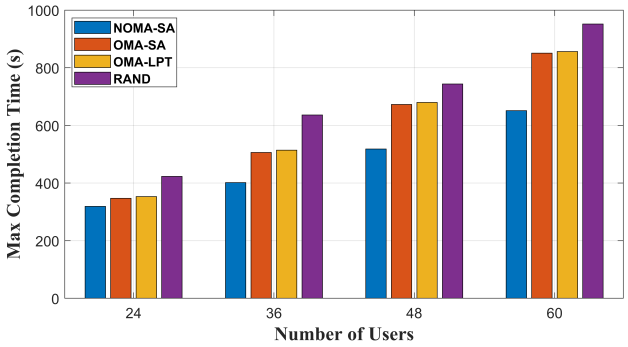

Fig. 7. Makespan versus the number of users ($F = 8$).

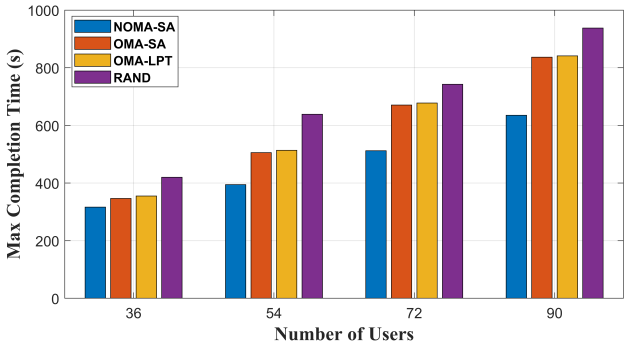

Fig. 8. Makespan versus the number of users ($F = 12$).

the SA loop 2 for scheduling optimization is performed. As shown by the figures, the NOMA scheme optimized by our SA-based serial collaborative method outperforms the conventional OMA scheme in terms of the makespan. Specifically, in Fig. 8, the proposed NOMA approach reduces the makespan by about 10% as compared to OMA-SA when $K = 36$. Moreover, given a fixed amount of frequency channels, the system performance gain of NOMA over OMA in terms of the maximum completion time increases significantly with the growing number of users. This can be demonstrated again by Fig. 8, where the benefit of NOMA against OMA-SA increases significantly from 10% to about 32% when $K = 90$. A possible reason is that with the increase of concurrent users,

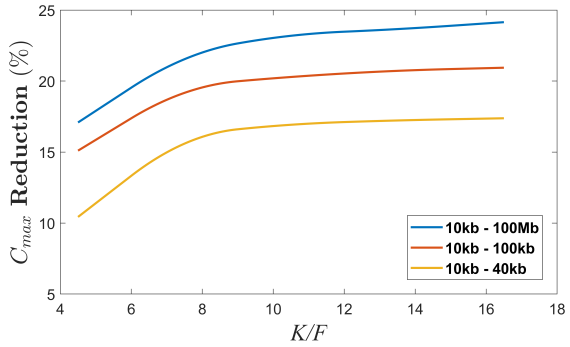

Fig. 9. Reduction on $C_{\max}$ achieved by NOMA using SA as compared to OMA-SA versus K/F ratio under different allowable package length variations, where $F = 4$.

the channel profiles are more diverse, which allows more user grouping choices in NOMA that result in more efficient utilization of the limited spectral resources. The performance of NOMA with random user pairing is even worse than conventional OMA scheme. This result reveals the importance of proper user grouping on the system performance of NOMA-enabled network. In addition, the performance gap between SA algorithm and LPT rule for OMA scheduling is about 2%, showing that the main performance gain of NOMA against OMA comes from the user clustering.

Fig. 9 shows the reduction in makespan by applying the proposed NOMA SA approach as compared to OMA-SA, where $F = 4$ and the packet size of each user varies within different ranges. We see that the benefit of NOMA against OMA increases as the number of users in the system increases. Moreover, the benefit of NOMA is more significant when the packet size variation is larger. Specifically, the reduction in makespan due to NOMA is about 16% when $K/F = 16$ and the packet size varies from 10^4 bits to 4×10^4 bits, but about 24% reduction in makespan is achieved when the packet size varies from 10^4 bits to 10^8 bits for the same K/F.

The number of iterations before the termination of Algorithm 1 for both SA loop 1 and SA loop 2 in Fig. 9 are recorded as follows. Specifically, when the packet size varies from 10^4 bits to 10^8 bits, SA loop 1 converges in a maximum of 15200 iterations, and SA loop 2 converges in a maximum of 17400 iterations. These two maximum iteration numbers increase to 26400 and 33200, respectively, when the packet size variation range is changed to $[10^4, 4{\times}10^4]$. Fig. 10 and Fig. 11 showcase the sum of processing time as evaluated in SA loop 1 and the maximum completion time as evaluated in SA loop 2 for a random network topology, respectively, where $K = 100$ and $F = 10$. We can see that the utility functions fluctuate but then converges as the number of iterations increases.

V. Conclusion

In this paper, we have proposed a SA-based serial collaborative optimization algorithm to minimize the maximum completion time of serving multiple uplink users that transmit packets of different lengths in a multi-channel NOMA system.

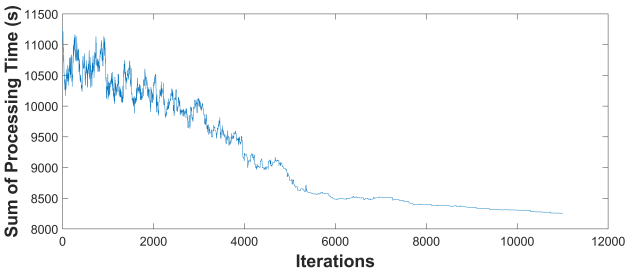

Fig. 10. Convergence result in SA loop 1.

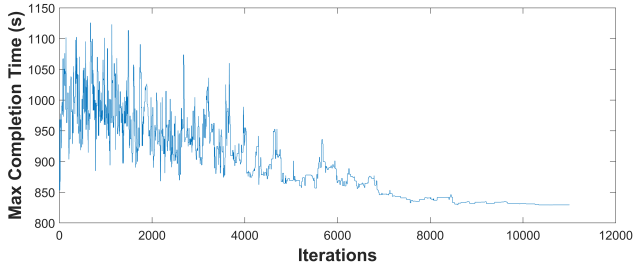

Fig. 11. Convergence result in SA loop 2.

We formulated the problem as a combinatorial optimization problem, analyzed its complexity and decomposed it into two separate parts. Simulation results have shown the convergence of the proposed algorithm. Moreover, the proposed algorithm is able to find clustering and scheduling solutions for NOMA that, on average, reduce the maximum completion time for more than 30% as compared to the OMA counterpart.

Appendix A
Proof of (8)

As shown in (7), the expression of $u_{i,j}$ has two forms. When $u_{i,j}$ takes the form of case 1,

$$\tau_i + \tau_j - u_{i,j} = \tau_i = \frac{L_i}{R_i^{\text{OMA}}} \geq 0. \tag{25}$$

When $u_{i,j}$ takes the form of case 2,

$$\tau_i + \tau_j - u_{i,j} = \tau_i - \frac{L_i}{R_i^{\text{OMA}}} + \frac{R_i^{\text{SIC},j}\tau_j}{R_i^{\text{OMA}}} = \frac{R_i^{\text{SIC},j}\tau_j}{R_i^{\text{OMA}}} \geq 0 \tag{26}$$

because $R_i^{\text{SIC},j}$, τ_j, and R_i^{OMA} are all non-negative. The proof completes since (25) and (26) prove (8).

Appendix B
A Lower Bound on the Stirling Number of the Second Kind

The number of ways of partitioning a set of n elements into k non-empty sets. Denoted by $S(n,k)$, Stirling numbers of the second kind obey the following recurrence relation:

$$S(n,k) = S(n-1,k-1) + kS(n-1,k). \tag{27}$$

We can obtain a lower bound for $S(n,k)$ as follows:

$$\begin{aligned} S(n,k) &= S(n-1,k-1) + kS(n-1,k) \\ &> kS(n-1,k) \\ &= k(S(n-2,k-1) + kS(n-2,k)) \\ &> k^2 S(n-2,k) \qquad (28) \\ &> k^{n-k} S(k,k) \qquad (29) \\ &= k^{n-k}, \qquad (30) \end{aligned}$$

where (28) follows because $S(n-2,k-1) \geq 0$, and (29) is obtained by repeating the argument that leads to (28).

References

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

[2] Z. Wei, L. Yang, D. W. K. Ng, J. Yuan, and L. Hanzo, "On the performance gain of NOMA over OMA in uplink communication systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 536–568, 2020.

[3] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, C.-L. I, and H. V. Poor, "Application of non-orthogonal multiple access in LTE and 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 185–191, 2017.

[4] H. Tabassum, M. S. Ali, E. Hossain, M. J. Hossain, and D. I. Kim, "Uplink vs. downlink NOMA in cellular networks: Challenges and research directions," in *IEEE VTC Spring*, 2017, pp. 1–7.

[5] Z. Ding, P. Fan, and H. V. Poor, "Impact of user pairing on 5G non-orthogonal multiple-access downlink transmissions," *IEEE Trans. on Vehicular Technology*, vol. 65, no. 8, pp. 6010–6023, 2016.

[6] I. Abu Mahady, E. Bedeer, S. Ikki, and H. Yanikomeroglu, "Sum-rate maximization of NOMA systems under imperfect successive interference cancellation," *IEEE Commun. Lett.*, vol. 23, no. 3, 2019.

[7] G. Jang, C. Lee, N. Kim, W. Kang, T.-V. Nguyen, and S. Cho, "Throughput maximization for uplink IoT NOMA systems with efficient resource allocation," in *Intl. Conf. on Inf. & Commun. Technol. Convergence*, 2020, pp. 1763–1765.

[8] S. Mishra, L. Salaün, C. W. Sung, and C. S. Chen, "Downlink connection density maximization for NB-IoT networks using NOMA with perfect and partial CSI," *IEEE Internet of Things Journal*, vol. 8, no. 14, 2021.

[9] O. Altintast, Y. Atsumi, and T. Yoshida, "On a packet scheduling mechanism for supporting delay sensitive applications on high speed networks," in *Intl. Conf. Inf. Commun. & Signal*, 1997, pp. 1441–1446.

[10] M. Mohsenivatani, Y. Liu, M. Derakhshani, S. Parsaeefard, and S. Lambotharan, "Completion-time-driven scheduling for uplink NOMA-enabled wireless networks," *IEEE Commun. Lett.*, vol. 24, no. 8, 2020.

[11] A. Wang, L. Lei, E. Lagunas, S. Chatzinotas, and B. Ottersten, "Completion time minimization in NOMA systems: Learning for combinatorial optimization," *IEEE Netw. Lett.*, vol. 3, no. 1, pp. 15–18, 2021.

[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[13] V. Granville, M. Krivanek, and J.-P. Rasson, "Simulated annealing: A proof of convergence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 652–656, 1994.

[14] H. Szu and R. Hartley, "Nonconvex optimization by fast simulated annealing," *Proc. IEEE*, vol. 75, no. 11, pp. 1538–1540, 1987.

[15] C. S. Chen, F. Baccelli, and L. Roullet, "Joint optimization of radio resources in small and macro cell networks," in *IEEE Vehicular Technology Conference (VTC Spring)*, 2011, pp. 1–5.

[16] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*. Addison Wesley, 1967.

[17] S. Ilie, D. Jeffrey, R. Corless, and X. Zhang, "Computation of Stirling numbers and generalizations," in *17th Intl. Symp. on Symbolic and Numeric Algorithms for Scientific Computing*, 2015, pp. 57–60.

[18] H. H. Hoos and C. Boutilier, "Solving combinatorial auctions using stochastic local search," in *17th National Conf. on Artificial Intelligence*. AAAI Press, 2000, p. 22–29.

[19] C. A. Glass and C. N. Potts, "A comparison of local search methods for flow shop scheduling," *Annals of Operations Research*, vol. 63, 1996.

[20] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311–329, 1988.

[21] X. Sun, L. Yu, and Y. Yang, "Jointly optimizing user clustering, power management, and wireless channel allocation for NOMA-based Internet of Things," *Digital Communications and Networks*, vol. 7, no. 1, 2021.