Minimum Delay Scheduling with Multi-Packet Transmission in Wireless Networks

Ali Abbasi and Majid Ghaderi
Department of Computer Science, University of Calgary
{aabbasi, mghaderi}@ucalgary.ca

***Abstract*—This paper studies the problem of minimum delay scheduling in wireless networks with multi-packet transmission capability. Specifically, we assume that the network employs superposition coding at the physical layer in order to implement multi-packet transmission. While most studies on superposition coding assume that unbounded number of packets can be coded together, physical and MAC layer limitations restrict the number of concurrent packets in a transmission set. Taking this constraint into consideration, we formulate the minimum delay scheduling as a combinatorial optimization problem and study its computational complexity under different transmission set sizes. We show that, when the transmission set size is limited to 2 packets, the problem can be solved optimally in polynomial time. Moreover, while the complexity of the problem for larger transmission set sizes is unknown, we present close-to-optimal heuristic algorithms that compute efficient solutions for the problem in polynomial time. Numerical results are also presented to study the efficiency and utility of the presented scheduling algorithms. Our results show that the heuristic algorithms are highly efficient, achieving delays that are less than 2% away from the optimal values.**

I. Introduction

In traditional wireless networks, a wireless device can only transmit or receive a single packet at a time. There are, however, advanced physical layer techniques such as successive interference cancellation (SIC) [1] and superposition coding (SC) [1] that enable, respectively, multiple packet reception and transmission at a device (even with a single antenna). With recent advances in multi-user signal processing, the implementation of multi-packet transmission and reception is advancing rapidly. Indeed, software-defined radio implementations of multi-packet reception and transmission have been reported in the literature [2], [3]. Such techniques increase the network capacity substantially by decoding the otherwise colliding packets.

However, appropriate scheduling algorithms are needed to coordinate transmissions properly in order to create optimal transmission opportunities for concurrent transmissions in the network. In practice, even with a single-packet-at-a-time scheduler at the link layer, a multi-packet capable physical layer results in some increase in the network throughput by preventing some of the potential collisions [2]. However, a multi-packet scheduling algorithm is required to fully exploit the potential of such a physical layer capability [4]. In this work, we investigate downlink scheduling in SC-enabled wireless networks, which is particularly important due to the dominance of downlink traffic in wireless networks.

A brief description of SC follows. Consider a system consisting of a transmitter and two receivers (*i.e.*, users) s_1 and s_2. Let h_1 and h_2 denote, respectively, the channel gain between the transmitter and s_1 and s_2. Without loss of generality, assume that $|h_1| \leq |h_2|$. The transmitter communicates with the receiver s_i with transmission power p_i and intends to send packet L_i to it for $i = 1, 2$. Using SC, the user with the low quality channel, *i.e.*, s_1, treats the other user's signal as noise and achieves the transmission rate of $R_1 = \log(1 + \frac{p_1|h_1|^2}{p_2|h_1|^2+N_0})$, where N_0 is the noise power at the receiver. Since s_2 enjoys a higher channel quality, it can also decode packet L_1 destined for s_1. After decoding L_1, s_2 reconstructs the corresponding analog signal and subtracts it from the combined received signal. This step is essentially an application of successive interference cancellation. The remaining signal is not affected by s_1's signal, therefore, L_2 is decoded without any interference from s_1. Thus the achieved transmission rate of s_2 is given by $R_2 = \log(1 + \frac{p_2|h_2|^2}{N_0})$. Comparing this rate with the traditional case where both users treat each others' signals as noise (s_2 achieves the rate $R_2 = \log(1 + \frac{p_2|h_2|^2}{p_1|h_2|^2+N_0})$ in the traditional case), SC obviously achieves a higher transmission rate.

While the above example demonstrates two levels of decoding, SC may utilize more than two levels to combine packets for multiple receivers. It has been shown that [5], in a system with n receivers, for any rate vector $\mathbf{r} = [r_1, \ldots, r_n]$ achievable via orthogonal division of resources (time or frequency), there exists a rate vector $\mathbf{r}' = [r'_1, \ldots, r'_n]$ achievable via SC in which $r' \geq r_i$ for all $1 \leq i \leq n$, where r_i is the transmission rate of receiver i. Note that SC exploits disparity of channels between the transmitter and receivers to provide a higher total transmission rate. It is more effective when channel gains across users are more diverse [1]. However when channel gains were equal, both orthogonalization and SC would result in the same transmission rates.

In this paper, we investigate the problem of minimum delay [1] Scheduling with Superposition Coding (SSC). The objective is to minimize the total delay spent to send all the users' data using multi-packet transmission capability of SC. Based on the optimality of SC compared to all orthogonalization methods, to attain this goal, all packets should be coded together and transmitted concurrently. However, practical considerations at the PHY and MAC layer limit the number of packets that can be coded together to a small constant. Considering this

[1]We use the terms delay and finish time interchangeably from now on.

constraint, the following questions need to be addressed in solving SSC:

1) How to partition packets to transmission sets in order to minimize the transmission time of all packets.
2) How to allocate power to packets in a transmission set in order to minimize the transmission time of the set.

We note that the latter question has been answered in the literature. The focus of this work, on the other hand, is on the former question.

There has been extensive research on resource allocation on broadcast channels [1], [6]. More recently, in [7], utility maximization through joint power and rate allocation in fading OFDMA broadcast channels is investigated. optimizing effective capacity of multicast service in wireless networks using SC is addressed in [8]. There has also been a line of works on scheduling with SC. For example, Eryilmaz *et al.* [9] proposed a queue proportional scheduling algorithm and demonstrated that it achieves the network capacity, while guaranteeing system stability. Seong *et al.* [10] extends the work by proposing an efficient geometric programming formulation for the optimal power allocation problem over broadcast channels. Inspiring for our work, the minimum delay region of the broadcast channel was characterized in [11]. The delay region is defined as the set of all delay vectors achievable by a scheduling algorithm when there are no further packet arrival in the system. Moreover, in [12] a practical design of a SC-enabled MAC scheduler was presented where the objective was to optimize throughput. An experimental implementation and evaluation of SC using software-define radio was presented in [13].

The main contributions of this paper can be summarized as follows:

- Minimum-delay scheduling problem is formulated formally.
- An optimal solution to a special instance of SSC where cardinality of transmission sets is limited to 2 is presented.
- To cope with the computational complexity of solving SSC for larger transmission sets, two polynomial-time heuristics algorithms are proposed.
- Numerical results are provided to demonstrate the efficiency of the proposed heuristics in solving SSC.

The rest of the paper is organized as follows. Section II describes the network and channel model used in the paper and formulates the problem. Optimal solution for a special case of the problem is described in Section II-D. In Section IV, heuristic algorithms are proposed to solve general instances of the problem. Sample numerical results are provided in Section V. Section VI concludes the paper.

II. System model

A. Network Model

We consider a wireless network consisting of a base station (access point) with maximum transmission power P and the set $\mathcal{S} = \{s_1, ..., s_n\}$ of receivers.

The base station sends signal x, which is composed of multiple user signals. The received signal at receiver s_l is given by:

$$y_l = h_l x + w_l,$$

where w_l is the Gaussian noise at the receiver, and h_l is the complex channel gain between the base station and receiver s_l. If the transmission power allocated for transmitting s_l's signal is p_l, then the received signal power at the receiver is given by $g_l p_l$, where $g_l = |h_l|^2$ is the corresponding power gain. In general, for the set of receivers $\{s_1, \ldots, s_K\}$, their allocated transmission powers $\{p_1, \ldots, p_K\}$, and power gains $\{g_1, \ldots, g_K\}$ such that $g_1 \leq \cdots \leq g_K$, the achievable rate region of SC is given by [1],

$$R_l \leq \log\big(1 + \frac{p_l g_l}{\sum_{j=l+1}^{K} p_j g_l + N_0}\big), \quad l \in \{1, \ldots, K\} \quad (1)$$

where N_0 is the noise power at a receiver.

Decoding transmitted information requires that both the base station and the receivers know the channel state information (CSI). To obtain this information, each receiver estimates its channel and feeds CSI to the base station via control channels. Also, the base station informs every receiver in a transmission set about the other users CSI at the beginning of each frame. We assume that changes in channel conditions are negligible during a scheduling frame.

B. Practical Considerations

Regardless of the objective *e.g.*, delay, throughput, capacity of the system can be achieved when SC is employed with suitable power vector [5]. However, there are practical constraints that limit the size of the transmission set, *i.e.*, the number of concurrently transmitted packets. Some of these limitations are summarized as follows.

- **Decoding complexity.** Decoding time at the receivers increases linearly with the size of the transmission set. To keep the decoding complexity and time manageable, only a few packets should be coded together.
- **Decoding error.** Removal of a signal from the composite signal cannot be done perfectly. Therefore, in practice, the size of the transmission set cannot be unbounded to limit the effect of imperfect signal cancellation at the receivers.
- **Minimum SINR requirement.** As the size of transmission set increases, the SINR level at the receivers decreases. Some decoders require a minimum SINR threshold to be able to decode the signal. Thus, transmission sets should be chosen small enough to satisfy the SINR threshold.

Due to these restrictions, we assume that transmission sets could have sizes of at most K where $K \ll |\mathcal{S}|$. Correspondingly, the *Constrained power set* of $\mathcal{S}$ denoted by $\mathcal{P}_K(\mathcal{S})$ is defined as $\mathcal{P}_K(\mathcal{S}) = \{\mathcal{S}_j \subseteq \mathcal{S}, |\mathcal{S}_j| \leq K\}$. Members of $\mathcal{P}_K(\mathcal{S})$ are the potential transmission sets that the scheduler can choose from.

C. Power Allocation

We assume that all the packets have the size of 1. For a given transmission set, the optimal delay can be attained when all packets are coded together. However, different power allocation policies could lead to different results. In the worst case, the total transmission power P could be allocated to just one receiver until its packet is sent. This strategy is effectively equal to orthogonalization of transmissions in time which offers no SC benefit. Given the increasingly ordered power gain vector $[g_1, \ldots, g_K]$ for the set $\mathcal{S}_j = \{s_1, \ldots, s_K\}$, we aim to find the power vector $[p_1, \ldots, p_K]$ that minimizes the delay for transmission of all users' information. Seong *et al.* [10] showed that the optimal power allocation ensures that all packet transmissions complete at the same time. When all packets have the same length, to obtain the same finish time, all transmission rates should be equal. Consequently, all signals attain the same SINR which leads to a simple method to find the optimal transmission powers. A backward substitution procedure is used starting from the receiver with the best channel. Suppose s^* is the maximum achievable SINR at each receiver. Then s^* is given by,

$$\frac{p_K g_K}{N_0} = s^* \Rightarrow p_K = \frac{N_0 s^*}{g_K}, \tag{2}$$

which yields,

$$\frac{p_{K-1} g_{K-1}}{p_K g_K + N_0} = s^* \Rightarrow p_{K-1} = \frac{N_0 s^*(1+s^*)}{g_{K-1}}\,. \tag{3}$$

In general, p_{K-i} is given by the following relation:

$$p_{K-i} = \frac{N_0 s^*(1+s^*)^i}{g_{K-i}}\,. \tag{4}$$

As $\sum_{l=1}^{K} p_l = P$, there are enough equations to find the optimal power allocation vector. The achievable transmission rate of the set $\mathcal{S}_j$ is then given by $R_{\mathcal{S}_j} = \log(1+s^*)$.

D. Problem definition

In this paper, we adopt the aggregate delay to transfer all users' information or *minimal potential delay* [14] as the objective. When the number of users is greater than K *i.e.*, $|\mathcal{S}| > K$, multiple transmission sets from $\mathcal{P}_K(\mathcal{S})$ should be selected to schedule all users.

Definition 1. Schedule: The set $\mathcal{E} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m | \mathcal{S}_j \in \mathcal{P}_K(\mathcal{S})\}$ is called a schedule if $\cup_{j=1}^{m} \mathcal{S}_j = \mathcal{S}$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for all $0 \leq i < j \leq m$.

We denote the set of all possible schedules by $\mathcal{E}_K(\mathcal{S})$. Delay of schedule $\mathcal{E}$ is defined as the summation of delay of all its subsets *i.e.*,

$$D(\mathcal{E}) = \sum_{\mathcal{S}_j \in \mathcal{E}} \frac{1}{R_{\mathcal{S}_j}}\,. \tag{5}$$

Accordingly, we aim to solve the following problem.

Definition 2. SSC: From the set $\mathcal{E}_K(\mathcal{S})$, find schedule $\mathcal{E}^*$ which results in minimal delay, *i.e.,*

$$\mathcal{E}^* = \underset{\mathcal{E} \in \mathcal{E}_K(\mathcal{S})}{\operatorname{argmin}} \quad D(\mathcal{E})\,. \tag{6}$$

III. Minimum Delay Scheduling Algorithms

Two cases of problem (8) where $K = 1$ and $K \geq |\mathcal{S}|$ are trivial to solve. In the former case which is *orthogonal scheduling*, packets are sent without using SC. In this case, the minimum delay is computed as follows

$$D_O(\mathcal{S}) = \sum_{s_l \in \mathcal{S}} \frac{1}{\log(1 + \frac{P g_l}{N_0})} \tag{7}$$

In the latter case, all packets are coded together and the problem is reduced to a simple power allocation as illustrated in Section II-C. However, the general case of $1 < K < |\mathcal{S}|$ is challenging. In the remainder of this section we present some algorithms to solve (8).

A. Integer programming solution

Problem (8) can be solved by standard integer programming algorithms such as branch and cut [15]. To do so, a binary decision variable x_j is defined for each member of $\mathcal{P}_K(\mathcal{S})$ as follows

$$x_j = \begin{cases} 1, & \mathcal{S}_j \text{ is present in the schedule} \\ 0, & \text{otherwise.} \end{cases}$$

Then, SSC is expressed as the following integer linear program

$$\begin{aligned} &\text{Minimize} \sum_{\mathcal{S}_j \in \mathcal{P}_K(\mathcal{S})} t_j x_j \\ &\text{subject to:} \sum_{s_l \in \mathcal{S}_j} x_j \geq 1, \quad \forall s_l \in \mathcal{S}\,. \end{aligned} \tag{8}$$

The objective function captures the desire to minimize the delay while the constraint enforces the coverage of all the receivers.

The number of variables in (8) is equal to the cardinality of $\mathcal{P}_K(\mathcal{S})$ which is given by, $|\mathcal{P}_K(\mathcal{S})| = \sum_{i=1}^{K} \binom{n}{i}$. Since K is considered to be a small constant, $K!$ is also a constant and $\binom{n}{i} \leq \binom{n}{K}$ for $i < K$, therefore,

$$\begin{aligned} |\mathcal{P}_K(\mathcal{S})| &= \sum_{i=1}^{K} \binom{n}{i} = \sum_{i=1}^{K} \frac{n(n-1)\ldots(n-i+1)}{i!} \leq K\binom{n}{K} \\ &\leq K\frac{n^K}{K!} \in O(n^K), \qquad K \ll n\,. \end{aligned}$$

Thus, the number of variables in (8) is polynomially bounded. However, the number of possible schedules $|\mathcal{E}_K(\mathcal{S})|$ is exponential in terms of $|\mathcal{S}|$. This can be realized based on the following argument. Let u_n denote the number of partitions of a set of size n to subsets of maximum size K. Based on the possibilities to choose the subset that contains receiver s_n , u_n can be computed by the following recurrence relation

$$u_n = \sum_{i=0}^{K-1} \binom{n-1}{i} u_{n-(i+1)},$$

where $\binom{n-1}{i}$ is the number of possible ways to choose i other members of that subset. The size of the subset excluding s_n

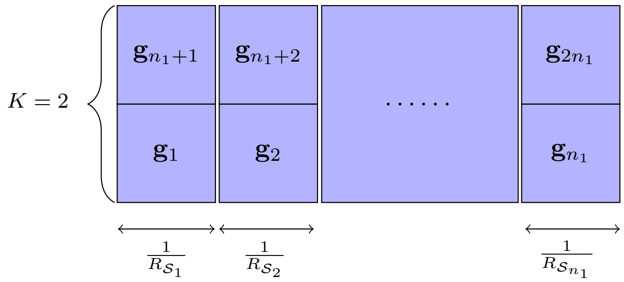

Fig. 1. SC scheduling where K = 2.

can vary from 0 to $K-1$. In the simplest form where $K = 2$, u_n is simplified to

$$\begin{aligned} u_n &= (n-1)u_{n-2} + u_{n-1} \\ &\geq u_{n-2} + u_{n-1}\,. \end{aligned}$$

A comparison between the recurrence relation for u_n and the one for Fibonacci numbers, *i.e.*, $f_n = f_{n-1} + f_{n-2}$, shows that the growth rate of u_n is faster than f_n, which we already know is exponential[2].

Thus, exhaustive search of all possible schedules to find the optimal one is a computationally intensive task. Interestingly, when the transmission set size is bounded to 2 ($K = 2$), under reasonable assumptions regarding the maximum achievable SINR during transmission of two messages, there exist exact polynomial-time algorithms to solve SSC which are presented next. We note that this case is important in practice as in most implementations, SC is limited to transmission of 2 packets [3], [12] due to aforementioned physical constraints.

B. Polynomial-time solution to SSC for $K = 2$

Suppose the number of receivers is even, that is $n = 2n_1$ for some $n_1 \geq 2$. Assume the associated set of power gains $\mathcal{G} = \{g_1, \ldots, g_{2n}\}$ are ordered increasingly, that is $g_i \leq g_j$ for all $i < j$. We define the *optimal configuration* for $\mathcal{G}$ as the schedule in which each $g_i, 1 \leq i \leq n_1$ is paired with g_{n_1+i}. Such a schedule is demonstrated in Figure 1. In the next two lemmas, we show that the optimal configuration is the optimal schedule when K is 2.

The next lemma demonstrates this case for $n_1 = 2$.

Lemma 1. Let $[g_1, g_2, g_3, g_4]$ denote an increasingly sorted list of power gains. Let $S(x, y)$ denote the maximum SINR that can be achieved simultaneously by two receivers with power gains x and y when total power P is distributed optimally between them. Assume that $S(g_3, g_4)\log(1 + S(g_3, g_4)) < 2$ and the maximum transmission set size is set to 2 *i.e.*, $K = 2$. Schedule $\mathcal{E}^* = \{\{g_1, g_3\}, \{g_2, g_4\}\}$ always results in minimum delay.

Proof. Other than $\mathcal{E}^*$, there are two other schedules namely, $\mathcal{E}_1 = \{\{g_1, g_4\}, \{g_2, g_3\}\}$ and $\mathcal{E}_2 = \{\{g_1, g_2\}, \{g_3, g_4\}\}$. Firstly, we demonstrate that the finish time of $\mathcal{E}^*$ is always lower than the finish time of $\mathcal{E}_1$.

[2] $f_n \geq r^{n-2}$ for $r = \frac{1+\sqrt{5}}{2}$ [16].

Case 1: $D(\mathcal{E}^*) < D(\mathcal{E}_1)$.

Let $I(,)$ denote the reciprocal of $S(,)$ *i.e.*, $I(x, y) = 1/S(x, y)$. Also let $R(,)$ and $T(,)$ denote the rate and time functions defined respectively as,

$$R(x, y) = \log(1 + S(x, y))$$

and

$$T(x, y) = \frac{1}{R(x, y)}\,.$$

We define function $F(.)$ as follows,

$$F(y) = T(g_1, y) - T(g_2, y), \quad \text{for} \quad y > g_2\,.$$

If $F(g_3) < F(g_4)$ *i.e.*, F is increasing, we have

$$T(g_1, g_3) - T(g_2, g_3) < T(g_1, g_4) - T(g_2, g_4),$$

and hence,

$$T(g_1, g_3) + T(g_2, g_4) < T(g_1, g_4) + T(g_2, g_3),$$

which shows that $\mathcal{E}^*$ results in a lower finish time in comparison to $\mathcal{E}_1$. To show that F is increasing, it is sufficient to demonstrate that its derivative is always positive. Derivative of F is given by,

$$\frac{dF(y)}{dy} = -\frac{\frac{\partial R(g_1,y)}{\partial y}}{R^2(g_1, y)} + \frac{\frac{\partial R(g_2,y)}{\partial y}}{R^2(g_2, y)}\,.$$

To show that $\frac{dF(y)}{dy} \geq 0$, the following inequality

$$\frac{\frac{\partial R(g_j,y)}{\partial y}}{R^2(g_j, y)} > \frac{\frac{\partial R(g_i,y)}{\partial y}}{R^2(g_i, y)},$$

has to be demonstrated which is equivalent to showing that $\frac{\frac{\partial R(x,y)}{\partial y}}{R^2(x,y)}$ is increasing in terms of x. Based on definitions of $T(,)$ and $R(,)$, this equals to proving that $\frac{\partial T(x,y)}{\partial x \partial y} < 0$.

Towards this goal, we initially establish a relation between $\frac{\partial T(x,y)}{\partial x \partial y}$ and $\frac{\partial I(x,y)}{\partial x \partial y}$. $\frac{\partial I(x,y)}{\partial y}$ is given by,

$$\frac{\partial I(x, y)}{\partial y} = -\frac{\frac{\partial S(x,y)}{\partial y}}{S^2(x, y)}\,.$$

Then, $\frac{\partial I(x,y)}{\partial x \partial y}$ is obtained as,

$$\begin{aligned} \frac{\partial I(x, y)}{\partial x \partial y} &= -\frac{\frac{\partial S(x,y)}{\partial x \partial y} S^2(x, y) - 2\frac{\partial S(x,y)}{\partial y}\frac{\partial S^2(x,y)}{\partial x}}{S(x, y)^4} \\ &= -\frac{\frac{\partial S(x,y)}{\partial x \partial y} S(x, y) - 2\frac{\partial S(x,y)}{\partial y}\frac{\partial S(x,y)}{\partial x}}{S(x, y)^3}\,. \end{aligned} \tag{9}$$

As SINR is always positive *i.e.*, $S(x, y) > 0$, $\frac{\partial I(x,y)}{\partial x \partial y}$ is negative if the numerator *i.e.*,

$$I_1(x, y) = \frac{\partial S(x, y)}{\partial x \partial y} S(x, y) - 2\frac{\partial S(x, y)}{\partial y}\frac{\partial S(x, y)}{\partial x}, \tag{10}$$

is positive or equivalently the following inequality

$$\frac{\frac{\partial S(x,y)}{\partial x \partial y}}{\frac{\partial S(x,y)}{\partial y}\frac{\partial S(x,y)}{\partial x}} > \frac{2}{S(x, y)}, \tag{11}$$

holds. As we will show later in the proof, (11) does actually hold. Repeating the same process for $T(,)$, $\frac{\partial T(x,y)}{\partial x \partial y}$ is found as follows,

$$\frac{\partial T(x,y)}{\partial x \partial y} = -\frac{1}{R^3(x,y)}\Big(-2\frac{\frac{\partial S(x,y)}{\partial x}}{1+S(x,y)}\frac{\frac{\partial S(x,y)}{\partial y}}{1+S(x,y)} + \frac{\frac{\partial S(x,y)}{\partial x \partial y}(1+S(x,y)) - \frac{\partial S(x,y)}{\partial x}\frac{\partial S(x,y)}{\partial y}}{(1+S(x,y))^2}R(x,y)\Big). \tag{12}$$

By removing the common terms in the numerator and denominator of (12), we find that $\frac{\partial T(x,y)}{\partial x \partial y} < 0$ only if,

$$T_1(x,y) = \frac{\partial S(x,y)}{\partial x \partial y} - \frac{\partial S(x,y)}{\partial x}\frac{\partial S(x,y)}{\partial y}\frac{R(x,y)+2}{1+S(x,y)}, \tag{13}$$

is positive. On the other hand, $T_1(x,y) > 0$ only if

$$\frac{\frac{\partial S(x,y)}{\partial x \partial y}}{\frac{\partial S(x,y)}{\partial y}\frac{\partial S(x,y)}{\partial x}} > \frac{R(x,y)+2}{1+S(x,y)}. \tag{14}$$

As g_3 and g_4 are the largest available power gains, assumption of the Lemma *i.e.*, $S(g_3,g_4)R(g_3,g_4) < 2$, implies that for all power gain pairs x and y in $\{g_1,g_2,g_3,g_4\}$, we also have

$$S(x,y)R(x,y) \leq 2. \tag{15}$$

(15) can be rewritten as follows,

$$\frac{2}{S(x,y)} > \frac{R(x,y)+2}{1+S(x,y)}, \tag{16}$$

which consists of the right-hand side terms of (11) and (14). Consequently, if $I_1(x,y) > 0$, (11) holds that based on (16) implies that (14) also holds which is the sufficient condition to conclude that $T_1(x,y) > 0$ or $\frac{\partial T(x,y)}{\partial x \partial y} < 0$. Therefore, to prove the first part of lemma, it remains to show that $I_1(x,y) > 0$ or $\frac{\partial I(x,y)}{\partial x \partial y} < 0$.

To this end, we obtain a closed-form representation of $S(,)$ in terms of the power gains and allocated powers to users. Based on (2), $S(x,y) = \frac{P_2 y}{N_0}$ where P_1 and P_2 are the optimal powers allocated to user s_1 and s_2. The following constraints hold on P_1 and P_2 (3):

$$\frac{P_1 x}{P_2 x + N_0} = \frac{P_2 y}{N_0} \quad \text{and} \quad P_1 + P_2 = P. \tag{17}$$

Solving the above equations, $S(,)$ is obtained as follows,

$$S(x,y) = \frac{-xN_0 - yN_0 + \sqrt{N_0\left(4Px^2y + (x+y)^2 N_0\right)}}{2xN_0}. \tag{18}$$

Accordingly, $\frac{\partial I(x,y)}{\partial x \partial y}$ is found as follows,

$$\frac{\partial I(x,y)}{\partial x \partial y} = \frac{(x-y)\left(N_0\left(2xyN_0 + y^2N_0 + x^2\left(4Py+N_0\right)\right)\right)^{3/2}}{\left(2xyN_0 + y^2N_0 + x^2\left(4Py+N_0\right)\right)^3} \tag{19}$$

which shows that $\frac{\partial I(x,y)}{\partial x \partial y}$ is always negative as $x < y$. This completes the proof for the optimality of the $\mathcal{E}^*$ compared to $\mathcal{E}_1$.

Case 2: $D(\mathcal{E}^*) < D(\mathcal{E}_2)$.

In this section, we demonstrate that the optimal configuration results in lower delay than $\mathcal{E}_2$. Let G denote the following function

$$G(x) = T(g_1,x) - T(x,g_4).$$

If $G(g_3) < G(g_2)$, then we have

$$T(g_1,g_3) - T(g_3,g_4) < T(g_1,g_2) - T(g_2,g_4),$$

and hence,

$$T(g_1,g_3) + T(g_2,g_4) < T(g_1,g_2) + T(g_3,g_4).$$

Therefore if $G(x)$ is shown to be decreasing, the result is achieved. To do so, we demonstrate that $\frac{dG(x)}{dx} < 0$ where $\frac{dG(x)}{dx}$ is given by,

$$\frac{dG(x)}{dx} = -\frac{\frac{\frac{\partial S(g_1,x)}{\partial x}}{1+S(g_1,x)}}{R^2(g_1,x)} + \frac{\frac{\frac{\partial S(x,g_4)}{\partial x}}{1+S(x,g_4)}}{R^2(x,g_4)} \tag{20}$$

The power gain pair $\{x,g_4\}$ achieves a higher SINR than the pair $\{g_1,x\}$ *i.e.*, $S(x,g_4) > S(g_1,x)$. Since the log function is monotonically increasing, this implies that,

$$\frac{1}{R^2(g_1,x)} > \frac{1}{R^2(x,g_4)}.$$

Therefore, showing the following inequality

$$\frac{\frac{\partial S(g_1,x)}{\partial x}}{1+S(g_1,x)} - \frac{\frac{\partial S(x,g_4)}{\partial x}}{1+S(x,g_4)} > 0, \tag{21}$$

for the numerators in (20), proves the result. Using the closed-form representation of $S(,)$ (18), we find (21) as,

$$\frac{\frac{-g_1-x+\sqrt{(ag_1)^2x+(g_1+x)^2}}{\sqrt{(ag_1)^2x+(g_1+x)^2}}}{2x} + \frac{\frac{2g_4\left((ax)^2+x+g_4-\sqrt{(ax)^2g_4+(x+g_4)^2}\right)}{\sqrt{(ax)^2g_4+(x+g_4)^2}\left(x-g_4+\sqrt{(ax)^2g_4+(x+g_4)^2}\right)}}{2x}, \tag{22}$$

where $a = 4P/N_0$. The first term of (22) is positive as

$$\sqrt{(ag_1)^2x+(g_1+x)^2} > g_1 + x.$$

In addition, we have

$$\left((ax)^2+x+g_4\right)^2 - \left(\sqrt{(ax)^2g_4+(x+g_4)^2}\right)^2 = (ax)^4 + 2(x+g_4)^2 + (ax)^2(2x+g_4),$$

which indicates that the second term is also positive. Therefore, (22) is positive which completes the proof.

□

Note that, the assumption regarding the maximum achievable SINR when transmitting two messages concurrently is not restrictive since the achievable SINR values using SC are typically low.

In the next lemma, we extend the previous lemma for the general case of $n_1 > 2$. Although it is not stated explicitly in the following we assume that any available pair of power gains x and y satisfies condition $S(x,y)R(x,y) < 2$.

Lemma 2. For $K = 2$ and $n = 2n_1$, the "optimal configuration" described above is the solution of SSC.

Proof. The lemma is proved by induction. The induction basis (case $n_1 = 2$) is proved in Lemma 1. Assume that the lemma holds for any set of power gains with cardinality $n = 2n' - 2$. To prove the induction step (case $n = 2n'$), we show that the optimal partition for the set $\mathcal{G} = \{g_1, \ldots, g_{2n'}\}$ contains the pair $\{g_{n'}, g_{2n'}\}$. If this is the case, the optimal schedule for the remaining set $\mathcal{G}\backslash\{g_{n'}, g_{2n'}\} = \{g_1, \ldots, g_{n'-1}, g_{n'+1}, \ldots, g_{2n'-1}\}$ is obtained based on the induction hypothesis as $\{\{g_1, g_{n'+1}\}, \ldots, \{g_{n'-1}, g_{2n'-1}\}\}$ which proves the result.

Suppose that instead of pairing up with $g_{n'}$, in the optimal partition, $g_{2n'}$ pairs with g_i where $g_i \neq g_{n'}$. If $g_i < g_{n'}$, according to the induction hypothesis $g_{n'}$ pairs with $g_{2n'-1}$ so the optimal solution contains the pairs $\{g_i, g_{2n'}\}$ and $\{g_{n'}, g_{2n'-1}\}$ where $i < n' < 2n' - 1 < 2n'$. However, according to Lemma 1, if these two pairs are replaced by $\{g_i, g_{2n'-1}\}$ and $\{g_{n'}, g_{2n'}\}$ a solution with lower delay is achieved . This contradicts the solution's optimality assumption. The same argument applies for the case where $g_i > g_{n'}$. In this case, according to the induction hypothesis, the solution contains $\{g_1, g_{n'}\}$ and $\{g_i, g_{2n'}\}$ where $1 < n' < i < 2n'$. Based on Lemma 1, a solution with lower delay is obtained if the two pairs are replaced by the pairs $\{g_1, g_i\}$ and $\{g_{n'}, g_{2n'}\}$. □

The complexity of finding the optimal configuration for a set of users is bounded by the complexity of the sort operation which is $O(n \log n)$. This result can be extended to finding the minimum delay schedule for any odd number of users, *i.e.*, $n = 2n_1 + 1$, as presented in the next lemma.

Lemma 3. For $K = 2$ and $n = 2n_1 + 1$, SSC can be solved optimally in $O(n^2)$.

Proof. Using the optimality property of SC [5], the optimal schedule would include n_1 pairs of packets that are coded and sent together and a packet that is sent individually. There are n possibilities to select the packet that will be transmitted without coding. For each of these possibilities, the delay is computed as the sum of the delay of transmuting the individual packet and the delay of the rest of packets. The latter one includes transmission of an even number of packets and can be obtained via Lemma 2. The optimal schedule is the one with the minimum delay among all n. Assuming that, the transmission time for a pair is computed in constant time, the complexity of finding the optimal schedule is $O(n^2)$ which proves the lemma. □

IV. Heuristic Algorithms

When $K > 2$, the optimal schedule does not conform a simple configuration. We conjecture that this case of the problem is NP-hard, however, computational complexity of SSC in this case is an open problem. Thus, in this section, we focus on heuristic algorithms of polynomial-time complexity to solve SSC.

A. *GreedySelect Algorithm*

We first present the GreedySelect algorithm. A greedy algorithm iteratively makes locally optimal decisions based on a greedy rule until a solution is found. Assume packets for subset $\mathcal{S}'$ of the receivers ($\mathcal{S}' \subset \mathcal{S}$) have already been transmitted. Suppose $\mathcal{S}_j$ is chosen to be transmitted next, then the optimal finish time for receivers in $\mathcal{S} - \mathcal{S}'$ is given by,

$$\frac{1}{R_{\mathcal{S}_j}} + D_{SC}(\mathcal{S}\backslash\mathcal{S}' \cup \mathcal{S}_j), \tag{23}$$

where $D_{SC}(\mathcal{S}\backslash\mathcal{S}' \cup \mathcal{S}_j)$ is the optimal delay for transmitting the remaining packets using SC. Then, the greedy rule for SSC is to choose the subset $\mathcal{S}_j \in \mathcal{P}_K(\mathcal{S}\backslash\mathcal{S}')$ that results in the minimum of (23). Since $D_{SC}(\mathcal{S}\backslash\mathcal{S}' \cup \mathcal{S}_j)$ is not known prior to scheduling all receives, it is approximated with the result obtained from orthogonal scheduling. Thus, we define the cost of subset $\mathcal{S}_j$ as follows:

$$\begin{aligned} \text{Cost}_j &= \frac{1}{R_{\mathcal{S}_j}} + D_O(\mathcal{S}\backslash\mathcal{S}' \cup \mathcal{S}_j) \\ &= \big[\frac{1}{R_{\mathcal{S}_j}} - D_O(\mathcal{S}_j)\big] + D_O(\mathcal{S}\backslash\mathcal{S}'). \end{aligned} \tag{24}$$

In (24), $D_O(\mathcal{S}\backslash\mathcal{S}')$ is constant, therefore, a minimum cost subset $\mathcal{S}_j$ is the one that offers the minimum $\frac{1}{R_{\mathcal{S}_j}} - D_O(\mathcal{S}_j)$. A greedy heuristic algorithm based on the definition of cost in (24) is presented in Algorithm 1. We show the list of selected subsets by $\mathcal{E}$ and the covered receivers by $\mathcal{S}'$. Initially both sets are empty. In each iteration of the while loop, the subset with the minimum cost is selected and added to the schedule $\mathcal{E}$. Moreover, the associated receivers are added to $\mathcal{S}'$. This process continues until all receivers are covered. There are $O(n^K)$ subsets to choose from. Assuming constant time to compute the transmission time for each subset, $O(n^K)$ takes to compute all the costs. At each step, at least one receiver is covered, therefore the running time of GreedySelect algorithm is in $O(n^{K+1})$.

Algorithm 1: GreedySelect Algorithm

Input: $\mathcal{S}, K$
Output: $\mathcal{E}$
begin
  $\mathcal{S}' \leftarrow \emptyset$;
  $\mathcal{E} \leftarrow \emptyset$;
  while $|\mathcal{S}'| < |\mathcal{S}|$ **do**
    $\mathcal{P} \leftarrow \mathcal{P}_K(\mathcal{S} - \mathcal{S}')$;
    $S_{min} \leftarrow \arg\min_{\mathcal{S}_j \in \mathcal{P}} \quad \frac{1}{R_{\mathcal{S}_j}} - D_O(\mathcal{S}_j)$;
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{S_{min}\}$;
    $\mathcal{S}' \leftarrow \mathcal{S}' \cup \mathcal{S}_{min}$;

The running time of GreedySelect algorithm depends on the maximum size of transmission set K. As a result, although GreedySelect features polynomial-time computational complexity, its running time of GreedySelect becomes prohibitive for larger Ks. In the next subsection, we introduce GreedyMerge algorithm to address this problem.

B. GreedyMerge Algorithm

In GreedySelect, the selected subset is chosen based on the approximation of finish time for a possibly large number of remaining users. Moreover, the decision at each step is final. Instead, in GreedyMerge algorithm, the transmission sets are greedily and incrementally constructed using a bottom-up process. The algorithm is demonstrated in Algorithm 2. The initial schedule consists of all individual users, similar to orthogonal transmission scenario. At each step of the algorithm, we merge the two transmission subsets that combining them gives the highest gain compared to orthogonal transmission, *i.e.*, the sets $\mathcal{S}_i$ and $\mathcal{S}_j$ such that

$$\frac{1}{R_{\mathcal{S}_i}} + \frac{1}{R_{\mathcal{S}_j}} - \frac{1}{R_{\mathcal{S}_i \cup \mathcal{S}_j}}$$

is maximal. The process continues until no further set union can be done. To find the two subsets that satisfy this condition, at most $\binom{n}{2} \in O(n^2)$ comparisons are required. Moreover, in each iteration of the algorithm, one transmission subset is removed from the schedule. Hence, the number of iterations is at most $n - n/K \in O(n)$. Overall, the computational complexity of the GreedyMerge algorithm is then in $O(n^3)$, which is independent of K.

Algorithm 2: GreedyMerge Algorithm

Input: $\mathcal{S}, K$
Output: $\mathcal{E}$
begin
- $\mathcal{S}_i \leftarrow \{s_i\}, \forall i, 1 \le i \le |\mathcal{S}|\}$;
- $\mathcal{E} \leftarrow \{\mathcal{S}_1 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{|\mathcal{S}|}\}$;
- $i_m \leftarrow 0$, $j_m \leftarrow 0$;
- **while** $i_m, j_m \neq 0$ **do**
  - $\mathcal{C} = \{(\mathcal{S}_i, \mathcal{S}_j), |\mathcal{S}_i \cup \mathcal{S}_j| \le K\}$;
  - $\mathcal{S}_{i_m}, \mathcal{S}_{j_m} \leftarrow \arg\max_{(\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{C}} \frac{1}{R_{\mathcal{S}_i}} + \frac{1}{R_{\mathcal{S}_j}} - \frac{1}{R_{\mathcal{S}_i \cup \mathcal{S}_j}}$;
  - **if** $\mathcal{S}_{i_m} \neq \emptyset$ **then**
    - $\mathcal{E} \leftarrow \mathcal{E} \cup \{\mathcal{S}_{i_m} \cup \mathcal{S}_{j_m}\}$;
    - $\mathcal{E} \leftarrow \mathcal{E} \backslash \mathcal{S}_{i_m}$;
    - $\mathcal{E} \leftarrow \mathcal{E} \backslash \mathcal{S}_{j_m}$;

V. Numerical results

In this section, we provide numerical results to compare different scheduling algorithms in various network configurations. The results are obtained using a simulator that is written in Python. Each result depicted in this section is obtained by averaging over 20 simulation runs with different seeds. Power allocation among users of a transmission set is formulated as a geometric program which is described next. The program is implemented using CVXOPT [17] package.

A. Geometric Program for Optimal Power Allocation

Allocation of power among users can be represented as the maximization of the minimum received SINR *i.e.*, $s = \min_i\{\frac{p_i g_i}{\sum_{j=i+1}^{n} p_j g_i + N_0}\}$ for all power gains $g_1, \ldots, g_n$. This objective can be expressed as follows

$$\begin{aligned} &\text{Maximize } s \\ &\text{subject to: } s \le \frac{p_i g_i}{\sum_{j=i+1}^{n} p_j g_i + N_0} \quad \forall i \in 1, ..., n \\ &\qquad \sum_{i=1}^{n} p_i \le P\,. \end{aligned}$$

The first constraint of the above program can be represented as

$$\sum_{j=i+1}^{n} s p_j g_i + s N_0 \le p_i g_i\,.$$

Due to term sp_j, this is a quadratic constraint which its associated second-order matrix is not positive semidefinite so the problem is nonconvex [18]. However s, p_i, g_i, P, and N_0 are all positive so the objective and constraints are posynomials and the problem can be transformed to the following geometric program [19],

$$\begin{aligned} &\text{Minimize } s^{-1} \\ &\text{subject to: } \sum_{j=i+1}^{n} s p_j p_i^{-1} + s p_i^{-1} g_i^{-1} N_0 \le 1 \quad \forall i \in 1, ..., n \\ &\qquad \sum_{i=1}^{n} \frac{p_i}{P} \le 1\,. \end{aligned}$$

With a logarithmic change of variables the problem is transformed to a convex problem which can be solved efficiently and reliably. At the optimal point, all instances of the first constraint of (25) become active and all users will receive the same SINR s^*.

B. Simulated Algorithms

In addition to GreedySelect and GreedyMerge algorithms, the following algorithms have been also implemented in our simulations:

- **Optimal**. This case represents coding all packets with SC without any bound on the size of the transmission sets.
- **ConstOptimal**. In this case, the size of the transmission sets is bounded. The scheduling problem formulated in (8) is solved numerically using Gurobi Optimizer [20] to find the optimal solution in this case.
- **Random**. In this algorithm, the number of transmission sets is determined to be $\lceil \frac{n}{K} \rceil$. For each user one of the transmission sets is selected randomly.
- **Orthogonal**. Messages are sent orthogonally and the finish time is computed as formulated in (7).

C. Simulation Parameters

The simulation parameters are adopted from [21] as the assumptions about the channel gains are consistent with the standard 3GPP propagation models. The power gain between the sender and a receiver is $g = f(d)$, where d is the distance from the sender to the receiver in (km). Also, $f(d) = 10^{h_0} d^{-\kappa}$ with path loss exponent $\kappa = 3.5$ and $h_0 = -14.4$. The background noise is $N_0 = -174$ dBm (Hz^{-1}). The bandwidth is 1 MHz and the maximum power P is selected to be 8W, 12W, or 16W. We simulate a base station that covers a circular area of radius 1 km. Users are uniformly distributed in the area. In our simulations, packet sizes are assumed to be equal and set to $L = 1$ kb.

D. Effect of Transmission Power

In the first set of simulations, we evaluate the effect of changing the maximum transmission power of the base station. Three power levels of 8W, 12W, and 16W are considered. The results are shown in Figure 2. We change the number of receivers from 6 to 15 and record the finish times obtained for all of the above algorithms. As depicted in the figure, a higher transmission power results in a lower finish time. In addition, as the number of users increases, the finish time increases as well. Figure 2 shows that SC methods constantly outperform the orthogonal algorithm by a large margin. Notably, $30-40\%$ reduction in the finish time is observed using a limited form of SC (*i.e.*, ConstOptimal) in comparison to the orthogonal scheduling, in all cases.

E. Effect of Transmission Set Size

The maximum transmission set size (K) is changed in this experiment to demonstrate its effect on the performance of SC scheduling. As shown in Figure V-E, decreasing K results in larger finish times. However, even when K is set to the minimum value of $K = 2$, the gain of using SC is substantial (about 27% improvement compared to the orthogonal scheduling). This is about 60% of the gain that can be obtained using SC without the bound on the transmission set size (which can achieve about 46% improvement compared to the orthogonal scheduling).

F. Performance of Greedy Heuristics

We have already observed in Figure 2 that the finish times obtained with ConstOptimal, GreedySelect, GreedyMerge algorithms are very close. To evaluate our heuristic algorithms, GreedySelect and GreedyMerge are compared with the random scheduling algorithm. As expected, the heuristic algorithms outperform the random algorithm in all cases. An interesting observation is that, while the results obtained from GreedySelect and GreedyMerge algorithms are almost the same, they are slightly in favor of GreedyMerge. This can be related to the fact that GreedyMerge works on smaller transmission sets to build larger ones. As a result, at each step of the algorithm, it makes smaller moves towards minimizing the finish time, which results in a slight improvement in its performance.

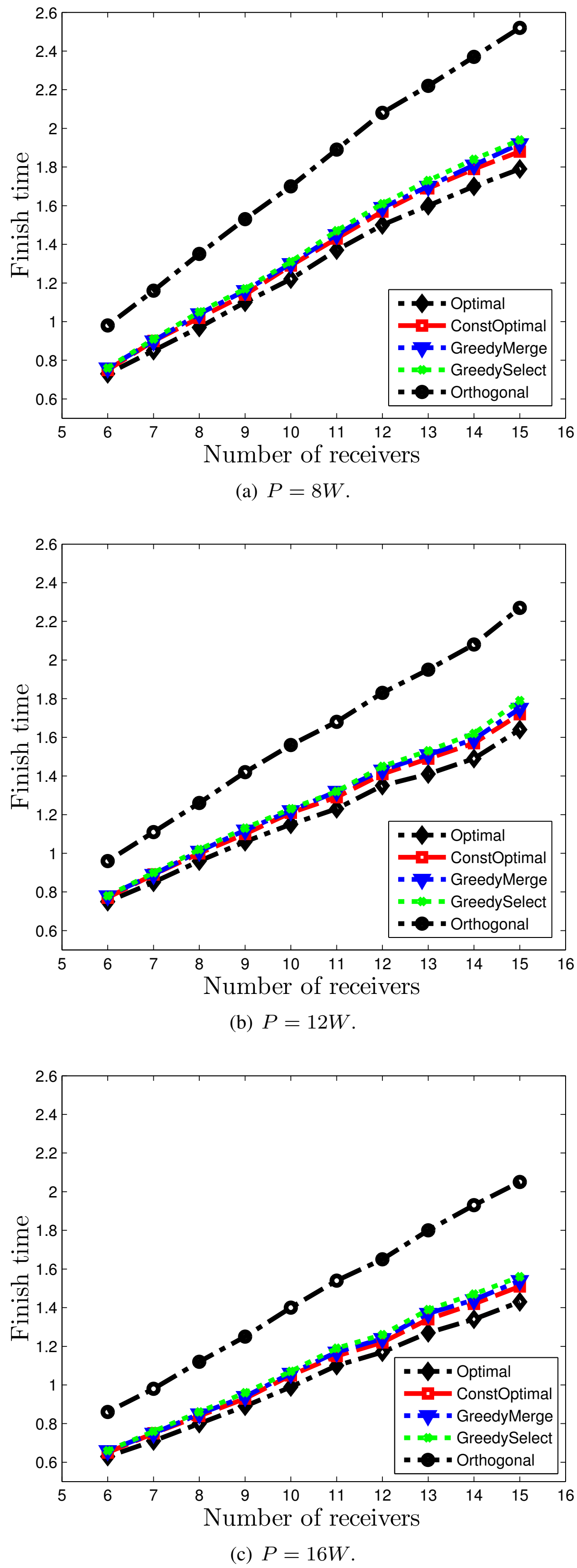

Fig. 2. Finish time with changing transmission power, $K = 3$. As demonstrated, a higher transmission power results in a lower finish time. In addition, even with limited form of SC (K is set to 3), $30-40\%$ reduction in the finish time is observed.

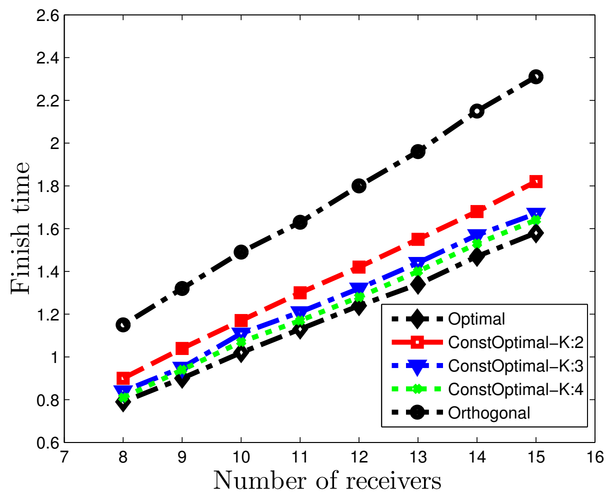

Fig. 3. Finish time with changing transmission set size, $P = 8W$. Larger transmission set size results in smaller finish times. Around 60% of the gain that can be obtained using unbounded SC compared to orthogonal transmission is also achievable when SC is used with bounded transmission set size of $K = 2$.

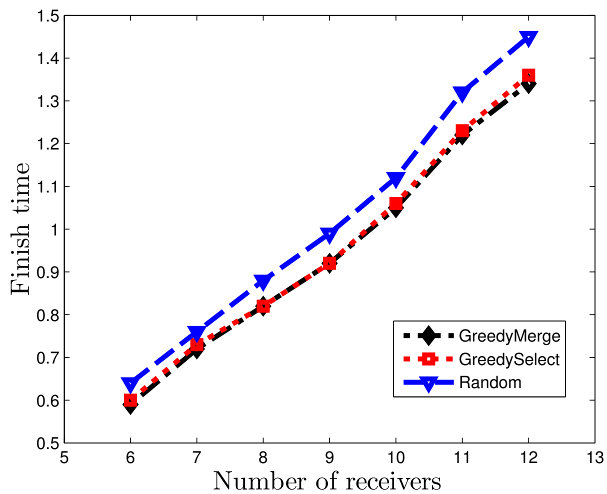

Fig. 4. Finish time of different heuristic algorithms, $P = 8W$, $K = 3$. Finish times obtained from GreedySelect and GreedyMerge are compared with the finish times achieved from the random algorithm. The GreedyMerge works better than the GreedySelect algorithm while both outperform the Random algorithm considerably.

VI. Conclusion and Future Works

The problem of minimum delay (in terms of the finish time) scheduling with superposition coding is investigated in this paper. Considering physical layer constraints, a simple framework to model SSC as an optimization problem is provided and complexity of the problem is discussed. Several algorithms are presented to solve SSC for different transmission set sizes. Through simulations, it is shown that the presented algorithms are able to improve finish time between 27-40% compared with the traditional orthogonal scheduling. Several issues are considered as possible future works. First, existence of polynomial-time algorithms to solve the problem when K is larger than 2 is unknown. However, we conjecture that this case is NP-hard. Moreover, we believe that the upper bound that is set on the SINR values to conform to the optimal configuration is unnecessary and the algorithm is correct for all SINR values. However, proving the result for general SINR's seems to be challenging as the closed-form of the achievable SINR is complex. In addition, in this work, we considered that communication between the base station and receivers takes place on a single channel. In the future, we plan to extend this work to parallel channels, where frequency bandwidth is divided into many subchannels. In this case, each user could have different power gain on different subchannels.

References

[1] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, 2005.

[2] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: interference cancellation for wireless LANs," in *Proc. ACM MobiCom*, San Francisco, USA, Sep 2008, pp. 339–350.

[3] S. Vanka, S. Srinivasa, Z. Gong, P. Vizi, K. Stamatiou, and M. Haenggi, "Superposition coding strategies: Design and experimental evaluation," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2628–2639, 2012.

[4] M. Mollanoori and M. Ghaderi, "On the performance of successive interference cancellation in random access networks," in *Proc. IEEE SECON*, Seoual, Korea, jun 2012.

[5] P. Bergmans and T. Cover, "Cooperative broadcasting," *IEEE Trans. Inf. Theory*, vol. 20, no. 3, pp. 317–324, May 1974.

[6] N. Jindal and A. Goldsmith, "Capacity and optimal power allocation for fading broadcast channels with minimum rates," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2895–2909, 2003.

[7] Z. Li and X. Wang, "Utility maximization over ergodic capacity regions of fading ofdma channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2478–2485, 2012.

[8] Q. Du and X. Zhang, "Effective capacity of superposition coding based mobile multicast in wireless networks," in *Proc. IEEE ICC*, DresDen, Germany, Jun 2009, pp. 1–5.

[9] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Throughput-optimal scheduling for broadcast channels," in *Proc. SPIE*, Denver, USA, August 2001, pp. 70–78.

[10] K. Seong, R. Narasimhan, and J. M. Cioffi, "Queue proportional scheduling via geometric programming in fading broadcast channels," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1593–1602, 2006.

[11] W. Lee, K. Seong, and J. M. Cioffi, "Optimal delay region for cross-layer resource allocation," in *Proc. IEEE GLOBECOM*, Washington, USA, Nov 2007, pp. 3343–3347.

[12] L. Li, R. Alimi, R. R. J. Shi, Y. Sun, H. Viswanathan, and Y. Yang, "Superposition coding for wireless mesh networks," in *Proc. ACM MobiCom*, Montreal, Canada, Sep 2007, pp. 330–333.

[13] P. Vizi, S. Vanka, S. Srinivasa, M. Haenggi, and Z. Gong, "Scheduling using superposition coding: Design and software radio implementation," in *Proc. IEEE RWS*, Phoenix, USA, Jan 2011, pp. 154–157.

[14] L. Massoulié and J. Roberts, "Bandwidth Sharing: Objectives and Algorithms," in *Proc. IEEE INFOCOM*, New York, USA, Mar 1999, pp. 1395–1403.

[15] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. Wiley New York, 1988.

[16] R. P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*. Pearson, 1988.

[17] J. Dahl and L. Vandenberghe, "CVXOPT: A python package for convex optimization," http://http://abel.ee.ucla.edu/cvxopt/, 2008.

[18] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Press, 2004.

[19] S. Boyd, S. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization and Engineering*, vol. 8, no. 1, pp. 67–127, 2007.

[20] "Gurobi optimizer," http://www.gurobi.com/.

[21] S. Borst, M. Markakis, and I. Saniee, "Distributed power allocation and user assignment in ofdma cellular networks," in *In Proc. Allerton Conference on Communication, Control, and Computing*, Urbana, USA, Sep 2011, pp. 1055–1063.