ARTICLE IN PRESS

Performance Evaluation ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms

Guy Grebla [a,*], Berk Birand [a], Peter van de Ven [b], Gil Zussman [a]

[a] Department of Electrical Engineering, Columbia University, New York, NY 10027, United States

[b] CWI, Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Coordinated Multi-Point (CoMP)
Joint transmission
Cellular networks
Approximation algorithms
Scheduling
Queueing networks

ABSTRACT

Due to the current trend towards smaller cells, an increasing number of users of cellular networks reside at the edge between two cells; these users typically receive poor service as a result of the relatively weak signal and strong interference. Coordinated Multi-Point (CoMP) with Joint Transmission (JT) is a cellular networking technique allowing multiple Base Stations (BSs) to jointly transmit to a single user. This improves the users' reception quality and facilitates better service to cell-edge users. We consider a CoMP-enabled network, comprised of multiple BSs interconnected via a backhaul network. We formulate the OFDMA Joint Scheduling (OJS) problem of determining a subframe schedule and deciding if and how to use JT in order to maximize some utility function. We show that the OJS problem is NP-hard. We develop optimal and approximation algorithms for specific and general topologies, respectively. We consider a time dimension and study a queueing model with packet arrivals in which the service rates for each subframe are obtained by solving the OJS problem. We prove that when the problem is formulated with a specific utility function and solved optimally in each subframe, the resulting scheduling policy is throughput-optimal. Via extensive simulations we show that the bulk of the gains from CoMP with JT can be achieved with low capacity backhaul. Moreover, our algorithms distribute the network resources evenly, increasing the inter-cell users' throughput at only a slight cost to the intra-cell users. This is the first step towards a rigorous, network-level understanding of the impact of cross-layer scheduling algorithms on CoMP networks.

1. Introduction

Cellular networks face an ever-increasing bandwidth demand, driven by the advent of sophisticated mobile devices and new applications. Satisfying this demand calls for improvements in the spectral utilization and reductions in inter-cell interference. The latter is becoming more relevant as the number of inter-cell users increases with ever-decreasing cell sizes. Such users are often unable to receive any transmission due to the high interference. Interference reduction can be efficiently accomplished through multi-cell coordination, known as Coordinated Multi-Point (CoMP) or Network-MIMO. One particularly promising category is CoMP with Joint Transmission (JT), where multiple Base Stations (BSs) jointly transmit to a single user, using the same time–frequency slots. This technique is incorporated in the LTE-Advanced standard [1]. Recently, CoMP with JT was shown to obtain substantial throughput gains in both indoor and outdoor testbeds [2].[1]

* Corresponding author.
E-mail address: guy@ee.columbia.edu (G. Grebla).

[1] There are two flavors of CoMP with JT: coherent [2] and non-coherent. We consider coherent JT but remark that all results can be directly extended to non-coherent JT.

http://dx.doi.org/10.1016/j.peva.2015.06.004
0166-5316/

As a result of the implementation of CoMP with JT in the LTE-Advanced standard, algorithm design and performance evaluation for these systems have recently received increased attention in the research literature (see, e.g., [3–5]). However, most existing work is concerned with developing heuristics designed for saturation conditions. In contrast, we consider a cellular network where new packets are generated over time, and construct a rigorous framework to develop scheduling algorithms for CoMP with JT that maximize throughput. This is achieved via a cross-layer approach, consisting of PHY (considering SINR-based transmission probabilities), MAC (deciding on a transmission schedule), and network layer (forwarding traffic over the backhaul).

We consider a cellular network comprised of multiple BSs interconnected via backhaul links. Users are assigned a serving and a secondary BS, and packets destined for a user can be transmitted either by the serving BS only, or jointly by the serving and secondary BSs. The latter provides better signal-to-interference-plus-noise ratio (SINR), but requires a subframe in both BSs, as well as forwarding the packet from the serving to the secondary BS prior to the transmission. A scheduling algorithm for CoMP with JT therefore needs to balance the performance benefits of transmitting packets using JT with the additional resources required for doing so.

We first focus on a single subframe, and study the OFDMA Joint Scheduling (OJS) problem of determining a schedule to maximize some utility function, given a set of packets for each user. Such a schedule determines which packets should be forwarded over the backhaul and which packets should be transmitted wirelessly, either using JT or by the serving BS only. We show that the OJS problem is NP-hard and describe a framework for solving it efficiently by decomposing it into problems related to knapsack and coloring. This allows us to develop an efficient algorithm for solving the OJS problem in bipartite backhaul network graphs. While backhaul network graphs are not necessarily bipartite, this result enables us to develop approximation algorithms for general backhaul graphs.

We then consider the network evolution over multiple subframes. We define a queueing model where the users are fixed, and packets for the various users are generated over time. Departures are determined by the schedule obtained from solving the OJS problem in each subframe. We characterize the capacity region (i.e., the packet arrival rates that can be sustained). Moreover, we demonstrate that when the OJS problem is formulated with a specific queue-length based utility function and solved optimally in each subframe, we obtain a MaxWeight-like scheduling policy (e.g., [6]), which we show to be throughput optimal. This is surprising, given the difference between OJS and the typical matching-type problems where MaxWeight scheduling performs well. Based on the queueing model, we present extensive simulation results to evaluate the performance of the proposed scheduling algorithms, as well as the benefits of JT. In particular, we consider different network topologies with an SINR-based channel model. We show that the bulk of the gains from CoMP with JT can be achieved with low capacity backhaul links. This result is highly relevant as the deployments of advanced cellular wireless technologies have a strong impact on mobile backhaul operational expenditure (OPEX), which amount to 20%–40% of total mobile operator's OPEX due to their reliance on T1/E1 backhaul copper lines [7]. A promising alternative is wireless backhaul (e.g., satellite, microwave), which is becoming a viable technology for geographically challenging regions and 5G networks. However, since such technology has limited capacity (due to, e.g., limited wireless spectrum and poor wireless channel conditions), our result are relevant to both present and future networks. We show that our algorithms distribute the network resources more evenly as the backhaul capacity increases. In fact, they increase the inter-cell users' throughput at only a slight cost to intra-cell users.

The main contributions of this paper are two-fold: (i) we define a rigorous model for CoMP with JT and develop novel scheduling algorithms with throughput guarantees for networks with queueing dynamics; (ii) via extensive simulations, we observe the benefits of JT and the tradeoffs related to its implementation.

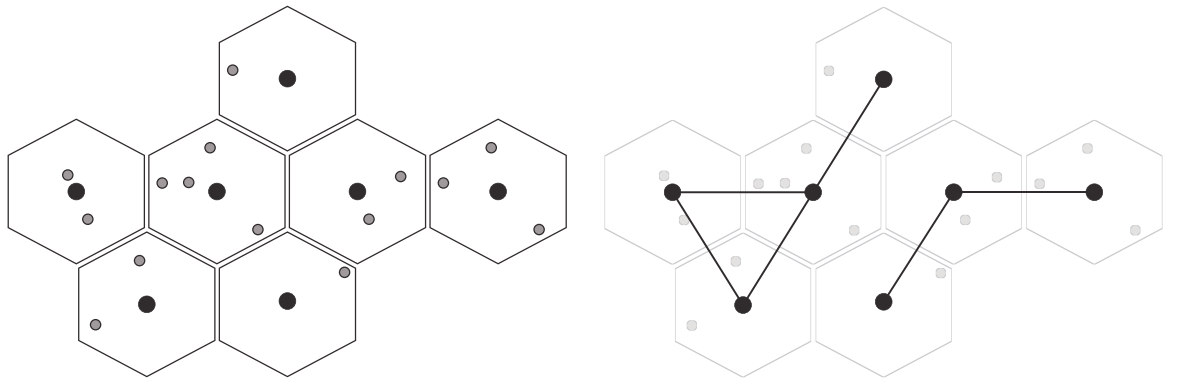
The rest of the paper is organized as follows. In Section 2 we discuss related work. In Section 3 we present the model. In Section 4, we introduce the OJS problem and show that it is NP-hard. We develop approximation algorithms for OJS in Section 5. In Section 6, we develop and present results for a queueing model, which we study through extensive simulation experiments in Section 7. Section 8 provides conclusions and directions for future research. All proofs can be found in the Appendix.

2. Related work

Previous work on scheduling for CoMP with JT has focused exclusively on analyzing the performance of heuristics, and has been limited mostly to networks that are saturated (i.e., have infinite backlog). The proposed heuristics are then evaluated via simulations or in testbeds (see, e.g., [2] and references therein). For example, [8–11,3] present heuristics for throughput maximization, assuming a perfect backhaul (infinite capacity and no delay), while [5] does the same for the finite backhaul case. In [4], the authors devise a heuristic for networks with perfect backhaul, and aim to minimize the backhaul traffic under certain SINR constraints. To our knowledge, [9,12] are the only studies that consider unsaturated networks, where traffic is generated over time rather than assuming an infinite supply of available packets. Both of these propose heuristics, assuming a perfect backhaul. The main contrast between our work and [12,10,11,3–5,9] is that we derive the first scheduling policies with performance guarantees. This is done for unsaturated networks, assuming finite backhaul capacity and positive delay over the backhaul.

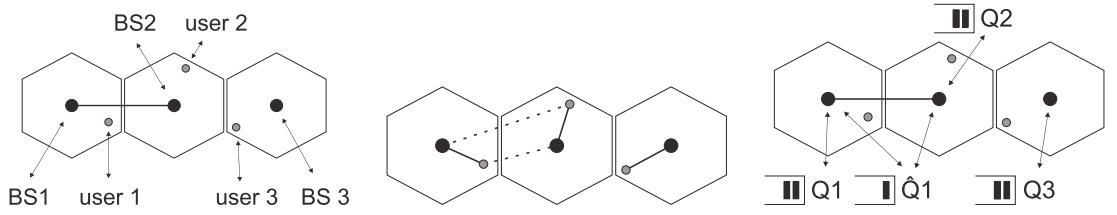
Models similar to the one considered in this paper have been investigated in the context of single-cell cellular transmissions. Packet-level scheduling algorithms for cellular networks are developed in [13,14]. In [15], approximation algorithms that provide queue stability are analyzed for a single BS.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004



(a) Wireless network. (b) Backhaul network and joint transmission graph.

Fig. 1. A cellular network comprised of basestations (black circles) and users (gray circles).



(a) Example network. (b) Primary (solid line) and secondary (dashed line) BSs. (c) The storage locations of the various queues.

Fig. 2. BS allocation and queues.

Closely-related technologies to CoMP with JT are network-MIMO, multi-cell MIMO, and multi-user MIMO (MU-MIMO) [16–19]. While theoretical studies (e.g., [18,20,21]) show that under certain conditions such technologies can completely cancel inter-cell interference, achieving these gains in *practical scenarios* is still challenging [19,17,16] (e.g., due to the high signal processing complexity). The Study of scheduling schemes for these technologies is subject to further research.

3. Network model

We consider an OFDMA cellular network comprised of a set of BSs $\mathcal{B} = \{1, \ldots, B\}$ and a fixed set of stationary users $\mathcal{N} = \{1, \ldots, N\}$, see Fig. 1(a). The BSs are connected by a backhaul network represented by a graph $G_J = (\mathcal{B}, \mathcal{C})$, where $\mathcal{C}$ is a set of backhaul links with $|\mathcal{C}| = C$. We refer to G_J as the *Joint Transmission Graph*, as only neighboring BSs in G_J can joint-transmit, see Fig. 1(b). We schedule over the downlink and assume that each backhaul link is bidirectional and that both directions share the link capacity, but remark that all results can be readily extended to directional backhaul links.

Definition 1. User n is associated with up to two BSs:

- The *serving BS* is denoted $\underset{\sim}{\mathrm{BS}}(n)$ and is defined as the BS that provides the highest SINR to user n.
- The *secondary BS* is denoted $\widehat{\mathrm{BS}}(n)$ and is defined as the BS with highest SINR that has a backhaul link to $\mathrm{BS}(n)$ in $\mathcal{C}$.

Note that some users may not have a secondary BS.

Packets destined for user n arrive at $\mathrm{BS}(n)$ and are stored in a queue Q_n. Transmission for user n can be either single-transmitted by $\mathrm{BS}(n)$ or joint-transmitted by $\mathrm{BS}(n)$ and $\widehat{\mathrm{BS}}(n)$. For a packet to be joint-transmitted, it first has to be forwarded over the backhaul to $\widehat{\mathrm{BS}}(n)$, and stored in a queue $\widehat{Q}_n$ for joint transmissions maintained at both $\mathrm{BS}(n)$ and $\widehat{\mathrm{BS}}(n)$. So a packet departs from Q_n when it is single-transmitted or forwarded across the backhaul, and a packet departs from $\widehat{Q}_n$ when it is joint-transmitted.

To illustrate this, consider the network in Fig. 2(a) with three users and three BSs. In Fig. 2(b) the primary and secondary BSs of each user are marked, with a solid and dashed line, respectively. Note that user 2's secondary BS is BS1 and not BS3, although the latter is closer, since BS3 does not have a backhaul link to BS2. User 3 does not have a secondary BS because its primary BS does not have any backhaul connections. Fig. 2(c) displays the various queues in play, and their locations.

Both wireless packet transmissions and forwarding over the backhaul lasts exactly a single subframe. Throughout, we assume that a central processing unit determines the schedule for all BSs based on perfect knowledge on the network state.

We consider a time-slotted model indexed by t, $t = 0, 1, \ldots$, where each time slot corresponds to a single subframe. Denote $L_n(t)$ and $\widehat{L}_n(t)$ the queue length of Q_n and $\widehat{Q}_n$ at the beginning of subframe t, respectively. Denote by $W_n(t)$ the

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

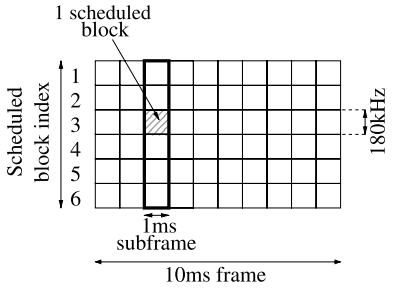


Fig. 3. Example of a frame (corresponding to 1.4 MHz LTE).

number of new packets generated for user n at the beginning of subframe t. The $W_n(t)$ are assumed to be i.i.d. over time, independent between users, and have finite second moment. We denote by $\mu_n^{(1)}(t)$, $\mu_n^{(2)}(t)$, and $\mu_n^{(3)}(t)$ the number of packets transmitted towards user n in subframe t using single and joint transmission, and the number of packets forwarded across the backhaul, respectively. These are determined by the resource allocation at each subframe, see Section 3.1 for more details. The evolution of the queue lengths can then be written as

$$L_n(t+1) = L_n(t) + W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t), \tag{1}$$

$$\widehat{L}_n(t+1) = \widehat{L}_n(t) + \mu_n^{(3)}(t) - \mu_n^{(2)}(t). \tag{2}$$

3.1. Subframe model

We now consider a single subframe consisting of scheduled blocks $\mathcal{S} = \{1, \ldots, S\}$ for each BS, see Fig. 3. In Sections 4 and 5 we discuss how to allocate resources within a single subframe, which determines the $\mu_n^{(j)}$, $j = 1, 2, 3$. The evolution of the queue lengths (1) and (2) over time is then discussed in Sections 6 and 7.

A packet i which is single-transmitted requires scheduled blocks in a subframe of BS($n(i)$), while a joint-transmitted packet requires scheduled blocks in the subframes of both BS($n(i)$) and $\widehat{\text{BS}}(n(i))$. In the latter case, the set of scheduled blocks used by BS($n(i)$) and $\widehat{\text{BS}}(n(i))$ *must have identical indices* since JT requires both BSs to transmit on the same scheduled blocks.

A packet i is characterized by the pair (n, β), where n is the receiving user, $\beta \in \{0, 1\}$ indicates whether a packet is in Q_n ($\beta = 0$) or in $\widehat{Q}_n$ ($\beta = 1$). The set of all packets is denoted $\mathcal{I}$, $|\mathcal{I}| = I$. Given a packet $i \in \mathcal{I}$, we denote by $n(i)$, $\beta(i)$ its user type and queue type, respectively.

When transmitted wirelessly, packet i is received successfully with probability $p(i)$. Additionally, the success probability $p(i)$ is independent of its allocated scheduled blocks, since we assume that the interference in all scheduled blocks is similar (due to frequency reuse 1). Note that if some scheduled blocks are unused, the interference is lower and better performance is obtained. We assume that $p(i)$ is higher if $\beta(i) = 1$ compared to $\beta(i) = 0$, since when using joint transmission the two BSs configure their transmission parameters such that the signal combines constructively at the user, resulting in greater SINR when a packet is joint-transmitted.

To simplify the presentation and due to space constraints, we make three assumptions: (i) forwarding a packet i over the backhaul is always successful; (ii) all packets are of same length and a packet transmission on a wireless channel requires one scheduled block from the subframe of each of its transmitting BSs; and (iii) the capacity of each backhaul link is K packets/subframe. In the accompanying technical report [22], we show that assumptions (i)–(iii) can be relaxed. Moreover, in [22] we show the applicability of our algorithms to the case where a packet can be transmitted using one of several Modulation and Coding Schemes (MCSs). We also remark that all the results in this paper can be readily applied to the setting with infinite backhaul capacity, by setting $K = S$. In our simulation study (Section 7) we evaluate our algorithms for the case where multiple MCSs are supported.

4. The OFDMA joint scheduling (OJS) problem

We now formulate the joint scheduling problem. In order to describe the BSs involved in each transmission, we introduce

$$h(i) = \begin{cases} \{\text{BS}(n(i))\} & \text{if } \beta(i) = 0, \\ \{\text{BS}(n(i)), \widehat{\text{BS}}(n(i))\} & \text{if } \beta(i) = 1. \end{cases}$$

If $\beta(i) = 0$ then h returns only the serving BS, and if $\beta(i) = 1$ it returns both the serving and secondary BS.

The function $u : \mathcal{I} \times \{0, 1\} \mapsto \mathbb{R}_+$ represents the utility of scheduling packet i over the backhaul ($u(i, 0)$) or wireless channel ($u(i, 1)$). Examples include throughput-based utility function u_T and fairness-based utility function u_F defined by

$$\begin{aligned} u_T(i, 0) &= \gamma, \qquad u_T(i, 1) = p(i), \\ u_F(i, 0) &= \gamma, \qquad u_F(i, 1) = \log p(i), \end{aligned} \tag{3}$$

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

where $\gamma > 0$ is a small constant that ensures packets are forwarded over the backhaul. Since here we consider a single-slot formulation, the utility of scheduling packets over the backhaul is not evident when the utility function is based only on wireless throughput; γ compensates for this. In Sections 6 and 7 we use a queue-length based utility function

$$\begin{aligned} u_Q(i,0) &= \max\{L_{n(i)} - \hat{L}_{n(i)}, 0\}, \\ u_Q(i,1) &= L_{n(i)} p(i), \end{aligned} \tag{4}$$

where L_n and $\hat{L}_n$ denote the queue length of Q_n and $\hat{Q}_n$, respectively. Our model and analysis can in fact handle a wide range of utility functions such as those used in [23].

Based on the set of packets $\mathcal{I}$ and the utility function u, the centralized scheduler determines the set of wireless transmissions to take place in the upcoming subframe, and what packets to forward over the backhaul.

The scheduler must also determine which scheduled blocks will be used for each packet transmission, such that for JT the scheduled blocks of the serving BS and secondary BS are aligned (i.e., have identical index). The scheduling decisions are represented using indicator variables $z_i \in \{0,1\}$, $y_i \in \{0,1\}$, and $x_{is} \in \{0,1\}$, where z_i indicates if packet i is transmitted wirelessly, y_i indicates if packet i is forwarded over the backhaul and x_{is} indicates if scheduled block s is used by packet i. The scheduler needs to solve the following integer programming problem (with $\boldsymbol{z} = (z_i)_{i\in\mathcal{I}}$, $\boldsymbol{y} = (y_i)_{i\in\mathcal{I}}$, and $\boldsymbol{x} = (x_{is})_{i\in\mathcal{I},\, s\in\mathcal{S}}$).

OFDMA Joint Scheduling (OJS) Problem:

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \sum_{i=1}^{I} z_i u(i,1) + y_i u(i,0) =: U(\boldsymbol{z},\boldsymbol{y})$$

$$\text{s.t. } z_i + y_i \le 1, \quad \forall\, i \in \mathcal{I}, \tag{5}$$

$$\sum_{\{i:a\in h(i)\}} z_i \le S, \quad \forall a \in \mathcal{B}; \qquad \sum_{\{i:h(i)=l\}} y_i \le K, \quad \forall l \in \mathcal{C}, \tag{6}$$

$$\sum_{s=1}^{S} x_{is} = z_i, \quad \forall i \in \mathcal{I};\ y_i = 0,\ \forall i \text{ s.t. } \beta(i) = 1, \tag{7}$$

$$\sum_{\{i:\ b\in h(i)\}} x_{is} \le 1, \quad \forall b \in \mathcal{B}\ \forall s \in \mathcal{S}, \tag{8}$$

$$z_i \in \{0,1\},\ y_i \in \{0,1\}, \quad \forall i \in \mathcal{I}, \tag{9}$$

$$x_{is} \in \{0,1\}, \quad \forall i \in \mathcal{I}\ \forall s \in \mathcal{S}. \tag{10}$$

Constraint (5) ensures a packet is scheduled at most once and resides in a single queue; (6) ensures capacities in each subframe and backhaul link are not exceeded; (7) ensures a scheduled block is allocated for each wireless transmission and packets in $\hat{Q}_n$ cannot be forwarded over the backhaul; and (8) ensures that each scheduled block is used at most once in the subframe of each BS.

To illustrate this problem, consider the network displayed in Fig. 4(a). This network comprises 3 BSs, 3 users and 7 packets, numbered $1,\ldots,7$. BS1 and BS2 are connected with a backhaul, so they can joint-transmit packets for user 1. Let $S = 2$, and assume that we want to allocate the scheduled block to achieve the following transmissions: (i) packet 3 should be joint-transmitted; (ii) packets 2, 4, 5 and 6 single-transmitted; and (iii) packet 1 forwarded over the backhaul from Q_1 to $\hat{Q}_1$. This schedule can be obtained with the assignment displayed in Fig. 4(b), and the following solution to the OJS problem:

$$\boldsymbol{z} = (0,1,1,1,1,1,0), \qquad \boldsymbol{y} = (1,0,0,0,0,0,0), \qquad \boldsymbol{x} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

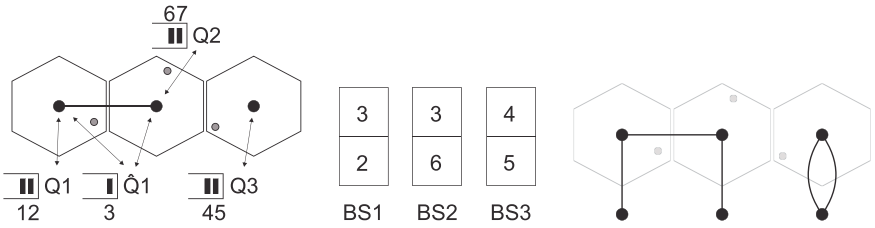
We now describe the complexity of the OJS problem.

Proposition 1. *OJS is strongly NP-hard even for instances in which all of the following hold:*

(a) $u(i,0) = 1, u(i,1) = 1, \forall i \in \mathcal{I}$;
(b) $L_n \le 1, \hat{L}_n \le 1, \forall n \in \mathcal{N}$.

The proof of Proposition 1 uses a reduction from the well-known problem of minimum edge coloring [24]. The reduction demonstrates that even for cases with sufficient bandwidth to accommodate all packet transmissions in the BSs subframes, obtaining a feasible schedule where joint-transmissions use an identical set of scheduled blocks is equivalent to the well-known problem of minimum edge coloring [24]. In Section 5, we use algorithms for minimum edge coloring when developing algorithms for OJS.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004



(a) A wireless network with 3 users and 7 packets (denoted 1…7).
(b) A possible allocation of the scheduled blocks.
(c) The resulting scheduled block graph.

Fig. 4. An example of a scheduled block allocation and the resulting scheduled block graph.

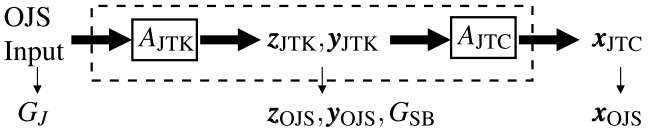


Fig. 5. The decomposition framework leading to algorithm $A_{\text{OJS}} = [A_{\text{JTK}}, A_{\text{JTC}}]$.

5. OJS problem—algorithms

In this section we develop algorithms to solve the OJS problem. First, we describe a framework for solving the OJS problem by decomposing it into problems related to knapsack and coloring, see Section 5.1. We then use this decomposition framework to develop efficient algorithms for OJS. In particular, in Section 5.2 we develop algorithms for instances where the joint transmission graph G_J (consisting of the BSs and the backhaul) is bipartite, in Section 5.3 we develop algorithms for instances where the joint transmission graph is planar and series–parallel, and in Section 5.4 we develop algorithms for general joint transmission graphs. Note that joint transmission graphs encountered in practice need not always be bipartite or planar and series–parallel. However, if this is the case, by using an algorithm that exploits the structure of the graph, we can guarantee lower computational complexity and more accurate results. The algorithms for the general case in Section 5.4 are based on those for bipartite graphs in Section 5.2.

We denote the approximation ratio of a given algorithm by α ($0 < \alpha \leq 1$). If the algorithm is optimal, we have $\alpha = 1$.

5.1. Decomposition framework

From Proposition 1 we see that, unless $P = NP$, an efficient optimal algorithm for general instances of OJS does not exist. In order to develop efficient approximations for the general case and optimal solutions for a subset of instances, we present two additional scheduling problems and explore their relation to OJS. These two problems are obtained by partitioning OJS into two parts, exploiting the fact that $\boldsymbol{x}$ only appears in (7), (8), and (10).

Joint Transmission Knapsack (JTK) Problem:

$$\max_{\boldsymbol{z},\boldsymbol{y}} \; U(\boldsymbol{z},\boldsymbol{y})$$

s.t. (5), (6), (9), $\exists \boldsymbol{x}:$ (7), (8), (10) hold.

Joint Transmission Coloring (JTC) Problem:

given $\boldsymbol{z}, \boldsymbol{y}$, find $\boldsymbol{x}$ s.t. (7), (8), (10) hold.

Note that the JTK problem resembles an assignment problem rather than a knapsack problem. We remark that this is due to' the assumptions that all packets have the same length and use a single MCS. We show in [22] that when relaxing these assumptions, JTK indeed generalizes to a knapsack-like problem.

We use A_{JTK} and A_{JTC} to denote an algorithm for JTK and JTC, respectively. A specific algorithm for problem P is denoted P–D where D identifies the algorithm. For instance, we write $A_{\text{JTK}} =$ JTK-GREEDY if we solve the JTK problem using a greedy algorithm. An instance of problem P consists of specific values for all variables in its constraints, except for $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{z}$ (JTK) and $\boldsymbol{x}$ (JTC).

The JTK problem differs from OJS in that it does not attempt to find $\boldsymbol{x}$ but guarantees that such $\boldsymbol{x}$ exists for its solution $\boldsymbol{z}, \boldsymbol{y}$. The JTC problem then finds $\boldsymbol{x}$ given $\boldsymbol{z}$ and $\boldsymbol{y}$, which is later shown to correspond to a coloring problem. It is ensured that if a solution exists for JTK, the corresponding JTC problem instance can also be solved. Thus we can decompose OJS by first solving JTK to obtain $\boldsymbol{z}, \boldsymbol{y}$ and then solving JTC to obtain $\boldsymbol{x}$, see Fig. 5 (some details in the figure are explained later). In Sections 5.2–5.4, we identify instances where the existence of a feasible $\boldsymbol{x}$ is guaranteed without the need to find $\boldsymbol{x}$, and use this to develop efficient algorithms for OJS.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

Table 1
The performance and input required for different algorithms $A_{\text{OJS}} = [A_{\text{JTK}}, A_{\text{JTC}}]$.

A_{JTK}	A_{JTC}	Ratio	Running time	Input G_J
JTK-MMK	JTC-BIP	α	$O(T_{\text{MMK}}(I, B, C, S))$	Bipartite
JTK-MAT	JTC-BIP	$\frac{2\alpha}{3\Delta(G_J)}$	$O(CT_{\text{MMK}}(I, 2, 1, S))$	Any
JTK-STA	JTC-BIP	$\frac{\alpha}{\Delta(G_J)}$	$O(T_{\text{MMK}}(I, \Delta(G_J)+1, \Delta(G_J), S)B^2)$	Any
JTK-PSP	JTC-PSP	α	$O(2^B + T_{\text{MMK}}(I, B+2^B, C, S))$	Planar ser. paral.

If algorithms A_{JTK} and A_{JTC} are used in this decomposition to solve JTK and JTC, respectively, we denote the corresponding OJS algorithms as $A_{\text{OJS}} = [A_{\text{JTK}}, A_{\text{JTC}}]$. As we shall demonstrate in this section, making this decomposition allows us to find efficient algorithms for solving OJS. Throughout the paper only optimal A_{JTC} algorithms are considered. The following lemma immediately follows.

Lemma 1. *If A_{JTK} is an α-approximation algorithm for the JTK problem and A_{JTC} an optimal algorithm for the JTC problem, the algorithm $A_{\text{OJS}} = [A_{\text{JTK}}, A_{\text{JTC}}]$ is an α-approximation for the OJS problem.*

We introduce the following definitions that will be used to solve JTC. Recall from Section 3 that $G_J = (\mathcal{B}, \mathcal{C})$ denotes the joint transmission graph that describes what BSs are connected by a backhaul link.

Definition 2. The *scheduled blocks graph* $G_{\text{SB}} = G_{\text{SB}}(G_J, \boldsymbol{z}) = (V_{\text{SB}}, E_{\text{SB}})$ of a JTC instance is defined by

$$V_{\text{SB}} = V_J \cup \{B+1, \ldots, 2B\}$$

and

$$E_{\text{SB}} = \Big\{\{\text{BS}(n(i)), \text{BS}(n(i)) + B\} \mid z_i = 1 \text{ and } \beta(i) = 0\Big\} \cup \Big\{\{\text{BS}(n(i)), \widehat{\text{BS}}(n(i))\} \mid z_i = 1 \text{ and } \beta(i) = 1\Big\}.$$

To interpret the definition of G_{SB}, recall that z_i denotes whether packet i does a wireless transmission ($z_i = 1$) or not ($z_i = 0$). So each edge in the scheduled block graph represents a wireless transmission, connecting the BSs that are engaged in the transmission. To avoid self-loops, we introduce dummy vertices $B+1, \ldots, 2B$ in case these BSs are involved in a single-transmission. Note that while G_J is a simple graph, G_{SB} need not be. As an example for G_J and G_{SB} consider the network depicted in Fig. 4(a). Using the solution described in the example at the end of Section 4, the resulting scheduled block graph is depicted in Fig. 4(c).

We now show that, using the scheduled block graph, JTC can be rewritten as an instance of the well-known edge-coloring (EC) problem [25], and solved accordingly. The input to the EC problem is a graph $G = (V, E)$ and the output is a coloring on the edges that uses a minimum number of colors.

Lemma 2. *JTC is equivalent to finding an edge coloring using at most S colors on $G_{\text{SB}} = (V_{\text{SB}}, E_{\text{SB}})$.*

As a consequence, JTC can be solved optimally by invoking an optimal algorithm A_{EC} for the EC problem on G_{SB}. Note that the EC problem is NP-hard.

Table 1 summarizes the different options we will describe in Sections 5.2–5.4 for solving OJS using the decomposition framework.

5.2. Algorithms for bipartite network graphs

We now develop algorithms for OJS instances in which G_J is bipartite. The results in this section can be used to solve such networks, and will provide the building blocks for the algorithms for general joint transmission graphs in Section 5.4. We start by describing an algorithm for JTK and an algorithm for JTC, and show how using them in the decomposition framework will result in an approximation algorithm for OJS. We require the following two lemmas. Denote by $\Delta(G)$ the maximum vertex degree of G.

Lemma 3. *If G_J is bipartite, then $G_{\text{SB}}(G_J, \boldsymbol{z})$ is bipartite for every $\boldsymbol{z}$.*

Lemma 4. *If G_{SB} is bipartite and $\Delta(G_{\text{SB}}) \le S$, then $\exists \boldsymbol{x}$ such that (7), (8), (10) hold.*

We now describe an algorithm for JTK based on the well-known Multidimensional Multiple-choice Knapsack (MMK) Problem [26]. Observing the formulation of MMK in [26], the input to MMK is a subset of the input to OJS. We define the algorithm A_{JTK} = JTK-MMK as simply running some algorithm A_{MMK} for solving MMK with a running time of T_{MMK} (for different A_{MMK} algorithms see Table 2) and show that it solves JTK for bipartite networks.

Lemma 5. *If G_J is bipartite and algorithm A_{MMK} is an α-approximation algorithm for MMK, $A_{\text{JTK}} = A_{\text{MMK}}$ is an α-approximation algorithm for JTK.*

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

Table 2
Algorithms for MMK.

A_{MMK}	Ratio	$T_{\text{MMK}}(I, B, C, S)$
DP [26]	Optimal	$O(S^{(B+C)}I(B+C))$
PTAS [27]	$1/(1+\epsilon)$	$O(I^{((B+C)/\epsilon)})$
Greedy [26]	∞	$O(I\log(I))$

Next, we describe an algorithm for JTC when the network graph is bipartite, by exploiting the connection to graph-coloring problems from Lemma 2. Let JTC-BIP be the edge coloring algorithm from [25]. Using Lemma 3, G_{SB} is bipartite and since also $\Delta(G_{\text{SB}}) \leq S$ it follows from [25] that JTC-BIP finds an edge coloring using at most S colors. Using Lemma 2 we conclude that JTC-BIP solves JTC. The running time of JTC-BIP is $O(|E_{\text{SB}}|\log\Delta(G_{\text{SB}})) = O(BS\log S)$. The following theorem is the main result of this section.

Theorem 1. *For bipartite networks, if A_{JTK} = JTK-MMK is an α-approximation for JTK, then A_{OJS} = [JTK-MMK, JTC-BIP] is an α-approximation for OJS.*

5.3. Algorithm for planar series–parallel graphs

We now develop an algorithm for OJS instances in which G_J is planar and series–parallel. We describe algorithms for JTK and JTC, and use them in the decomposition framework to devise an approximation algorithm for OJS. In this section we use similar ideas to those in Section 5.2. We need the following lemma whose proof is similar to that of Lemma 3:

Lemma 6. *If G_J is planar and series–parallel, G_{SB} is planar and series–parallel for every $\boldsymbol{z}$.*

We first describe Algorithm JTK-PSP that solves JTK when G_J is planar and series–parallel. The algorithm uses A_{MMK} to solve an MMK instance defined as follows. The number of dimensions is $D' = D + |\mathcal{B}_{\text{odd}}|$ where $\mathcal{B}_{\text{odd}} = \{\mathcal{B}' \subseteq \mathcal{B} : |\mathcal{B}'| \text{ is odd and } \geq 3\}$. The capacity for each new dimension associated with a set $\mathcal{B}' \in \mathcal{B}_{\text{odd}}$ is $S(|\mathcal{B}'|-1)/2$. The weight in each new dimension for each (i,r) such that $r > 0$ and $h(i,r) \subseteq \mathcal{B}'$ is set to $\Gamma(i,r)$; for all other cases it is set to zero. The algorithm concludes by scheduling packets for transmission according to the configurations selected in the solution returned by A_{MMK}.

We note that $|\mathcal{B}_{\text{odd}}| = O(2^B)$ and therefore in general Algorithm JTK-PSP may be impractical due to a very large running time. This algorithm is therefore more appropriate for small B. We now show that in some instances the running time can be improved. Note that if a set $\mathcal{B}' \in \mathcal{B}_{\text{odd}}$ has no more than $|\mathcal{B}'|-1$ edges in $\mathcal{C}$ that connects two nodes in $\mathcal{B}'$, it can be removed from $\mathcal{B}_{\text{odd}}$; if the number of such sets is large this can significantly decrease the running time.

For networks that are planar and series–parallel, JTC can be solved using the edge-coloring algorithm from [28]. We call this algorithm JTC-PSP, and note that its running time is $O(B\Delta(G_{\text{SB}})) = O(B\cdot S)$. The following theorem applies the decomposition framework to planar and series–parallel networks.

Theorem 2. *For planar and series–parallel networks, if JTK-MMK is an α-approximation for MMK then [JTK-PSP, JTC-PSP] is an α-approximation for OJS.*

5.4. Algorithms for general graphs

We now develop algorithms for general OJS instances, without imposing any conditions on G_J. We start with describing two approximation algorithms for JTK. For each approximation algorithm we show how using it in the decomposition framework will result in an approximation algorithm for OJS.

First, we describe Algorithm JTK-MAT which is based on computing a matching. For each $\{a,b\} \in \mathcal{C}$, the algorithm solves an instance of JTK defined by a network that has only two BSs a, b and the backhaul link with capacity K. Only packets that can be scheduled in such network are considered, and A_{MMK} with $B = 2$ and $C = 1$ is used to solve this limited instance. Each edge in $\mathcal{C}$ is assigned a weight equal to the total utility obtained when solving its limited JTK instance. Then, maximum weighted matching is found and the union of all solutions for edges in the matching is returned. This solution is feasible for the general JTK problem.

Theorem 3. *If algorithm A_{MMK} is an α-approximation algorithm, then A_{OJS} = [JTK-MAT(A_{MMK}), JTC-BIP] is a $(2\alpha)/(3\Delta(G_J))$-approximation algorithm for OJS.*

The maximum weight matching algorithm from [29] that takes $O(|E||V|\log|V|)$ time can be used in line 6; in this case the running time of Algorithm JTK-MAT is dominated by A_{MMK} in line 4. Therefore, the running time of Algorithm JTK-MAT is $O(CT_{\text{MMK}}(I,2,1,S))$, where T_{MMK} is the running time of A_{MMK} (see Table 2).

We now describe Algorithm JTK-STA which iterates over star subgraphs. It is similar to Algorithm JTK-MAT, but iterates over the vertices $b \in \mathcal{B}$ instead of the edges $e \in \mathcal{C}$. The approximation ratio of JTK-STA is better than that of JTK-MAT, but

Algorithm JTK-MAT Based on matching

1: **for** $b \in \mathcal{B}$ **do**
2: If b has no backhaul link, run A_{MMK} to solve a JTK instance with $\mathcal{B}' = \{b\}$, $\mathcal{I}' = \{i \in \mathcal{I} : h(i) = \mathcal{B}'\}$, $\mathcal{C}' = \{\}$ and set z_i and y_i as determined by A_{MMK}.
3: **for** $e = \{a, b\} \in \mathcal{C}$ **do**
4: Run A_{MMK} to solve a JTK instance with $\mathcal{B}' = \{a, b\}$, $\mathcal{I}' = \{i \in \mathcal{I} : h(i) \subseteq \mathcal{B}'\}$, $\mathcal{C}' = \{e\}$
5: Assign $U(\boldsymbol{z}, \boldsymbol{y})$ found by A_{MMK} as a weight for e
6: Compute maximum weight matching on G_J and store the result in the edge set $\mathcal{E}$
7: Set $z_i = 1$ ($y_i = 1$) only if $\exists e \in \mathcal{E}$ such that $z_i = 1$ ($y_i = 1$) in the solution returned in line 4 for edge e

its running time is worse. For each vertex b, a JTK instance is constructed using only b and its neighbors in G_J. The solution of this instance, $\boldsymbol{z}, \boldsymbol{y}$, is assigned to b. Next, the algorithm finds the vertex $b_{\max}$ associated with maximum total utility. If $\boldsymbol{z}_{\max}, \boldsymbol{y}_{\max}$ is the solution associated with $b_{\max}$, the algorithm schedules the packets indicated by $\boldsymbol{z}_{\max}, \boldsymbol{y}_{\max}$. The vertex $b_{\max}$ and its neighbors are removed from $\mathcal{B}$. This process is repeated until $\mathcal{B}$ is empty.

Note that after the first vertex is removed from $\mathcal{B}$, in order to update the weights it is sufficient to consider 2-hop neighbors of $b_{\max}$ in line 14, since weights of other vertices remain unchanged. The running time of JTK-STA is $O(T_{\text{MMK}}(I, \Delta(G_J) + 1, \Delta(G_J), S)B^2)$.

The following theorem proves that OJS can be solved approximately when JTK-STA is used in the decomposition framework.

Theorem 4. *If algorithm A_{MMK} is an α-approximation algorithm, then, $A_{\text{OJS}} = [$JTK-STA(A_{MMK}), JTC-BIP$]$ is an $(\alpha/\Delta(G_J))$-approximation algorithm for OJS.*

Algorithm JTK-STA Based on star subgraphs

1: **function** SOL-STAR(b, $\mathcal{B}'$, $\mathcal{C}'$, $\mathcal{I}'$)
2: Run A_{MMK} to solve a JTK instance defined with $\widetilde{\mathcal{B}}' = \{b\} \cup \{a : \{a, b\} \in \mathcal{C}'\}$, $\widetilde{\mathcal{C}}' = \{\{a, b\} : \{a, b\} \in \mathcal{C}',\ a \in \widetilde{\mathcal{B}}',\ b \in \widetilde{\mathcal{B}}'\}$, $\widetilde{\mathcal{I}}' = \{i \in \mathcal{I}' : h(i) \subseteq \widetilde{\mathcal{B}}'\}$
return $\boldsymbol{z}, \boldsymbol{y}$ as determined by A_{MMK}
3: **for** $b \in \mathcal{B}$ **do**
4: $\boldsymbol{Z}[b], \boldsymbol{Y}[b] \leftarrow$ SOL-STAR(b, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{I}$)
5: Assign $U(\boldsymbol{Z}[b], \boldsymbol{Y}[b])$ as a weight for b in $\mathcal{B}$
6: Initialize $\mathcal{B}'' \leftarrow \mathcal{B}$; $\mathcal{C}'' \leftarrow \mathcal{C}$; $\mathcal{I}'' \leftarrow \mathcal{I}$
7: **repeat**
8: Find the vertex $b_{\max}$ in $\mathcal{B}''$ with maximum weight
9: $\widetilde{\mathcal{I}} \leftarrow \{i \in \mathcal{I}'' : \boldsymbol{Z}[b_{\max}]_i + \boldsymbol{Y}[b_{\max}]_i = 1\}$
10: for all $i \in \widetilde{\mathcal{I}}$, $z''_i \leftarrow \boldsymbol{Z}[b_{\max}]_i$, $y''_i \leftarrow \boldsymbol{y}[b_{\max}]_i$
11: $\mathcal{J} \leftarrow \{a \in \mathcal{B}'' : \exists b \in \mathcal{B}'', \{b_{\max}, b\} \in \mathcal{C}'', \{a, b\} \in \mathcal{C}''\}$
12: $\mathcal{I}'' \leftarrow \mathcal{I}'' \setminus \widetilde{\mathcal{I}}$; $\mathcal{B}'' \leftarrow \mathcal{B}'' \setminus \{a : \{a, b_{\max}\} \in \mathcal{C}''\}$
13: Remove from $\mathcal{C}''$ edges with an endpoint not in $\mathcal{B}''$
14: **for** $b \in \mathcal{J}$ **do**
15: $\boldsymbol{Z}[b], \boldsymbol{Y}[b] \leftarrow$ SOL-STAR(b, $\mathcal{B}''$, $\mathcal{C}''$, $\mathcal{I}''$)
16: Update $U(\boldsymbol{Z}[b], \boldsymbol{Y}[b])$ as a weight for b in $\mathcal{B}''$
17: **until** $\mathcal{B}''$ is empty
18: **return** $\boldsymbol{z}''$

6. Queueing dynamics

The OJS problem discussed in Sections 4 and 5 schedules transmissions within a single subframe. We now expand the scope to multiple subframes, where packets arrive and depart over time, and study the evolution of the users' queues. Our objective is to identify a scheduling policy that is maximum stable (or throughput optimal). Under such a policy, the queue-length process is positive recurrent for any arrival for which a stabilizing policy exists, see, e.g., [30,6,31,32]. We prove that by using a specific utility function and an algorithm for solving the OJS problem (we refer to this combination as a scheduling policy), we obtain a MaxWeight-like scheduling policy (see, e.g., [6]), which is throughput optimal.

Let $\boldsymbol{L}(t) = (L_1(t), \widehat{L}_1(t), \ldots, L_n(t), \widehat{L}_n(t))$ denote the queue length process at time t, so $\{\boldsymbol{L}(t)\}_{t \geq 0}$ is the stochastic process that tracks the queue-length evolution over time. We solve the OJS problem in each subframe, given a certain utility function. Let us denote by $\boldsymbol{y}(t) = (y_i(t))_{i \in \mathcal{I}}$ and $\boldsymbol{z}(t) = (z_i(t))_{i \in \mathcal{I}}$ the solution of OJS in slot t. Here $(\boldsymbol{y}(t), \boldsymbol{z}(t))$ can represent both an exact solution or an approximation.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

Recall that $W_n(t)$ is the number of packets generated for user n at time t. Let $\lambda_n = \mathbb{E}\{W_n(0)\}$, and define $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$ to be the arrival rates. We denote by $\mu_n^{(1)}(t; \boldsymbol{z}(t))$, $\mu_n^{(2)}(t; \boldsymbol{z}(t))$, and $\mu_n^{(3)}(t; \boldsymbol{y}(t))$ the number of packets transmitted towards user n in subframe t using single and joint transmission, and the number of packets forwarded across the backhaul, respectively, given solution $(\boldsymbol{y}(t), \boldsymbol{z}(t))$ of OJS.

Denote $\mathcal{I}_n(t)$ the set of packets in Q_n at time t (so $|\mathcal{I}_n(t)| = L_n(t)$), and $\widehat{\mathcal{I}}_n(t)$ the set of packets in $\widehat{Q}_n$. The $\mu_n^{(j)}$, $j = 1, 2, 3$ can be written as

$$\mu_n^{(1)}(t; \boldsymbol{z}) = \sum_{i \in \mathcal{I}_n(t)} z_i(t) Y_{it}, \qquad \mu_n^{(2)}(t; \boldsymbol{z}) = \sum_{i \in \widehat{\mathcal{I}}_n(t)} z_i(t) Y_{it}, \qquad \mu_n^{(3)}(t; \boldsymbol{y}) = \sum_{i \in \mathcal{I}_n} y_i(t),$$

where the $Y_{it} \sim \text{Ber}(p(i))$ are mutually independent Bernoulli distributed random variables that represent whether packet transmissions are successful. For notational convenience, we write $\mu_n^{(j)}(t)$ to represent the transmission rates at time t.

It is readily seen that the joint queue-length process $\{\boldsymbol{L}(t)\}_{t \geq 0}$ is Markovian.

We now analyze the traffic intensity that can be sustained by the queueing system described by (1) and (2). The stability region of a particular policy is defined as the set of all arrival rates such that the $\{\boldsymbol{L}(t)\}_{t \geq 0}$ is positive recurrent is called the *stability region* of this particular policy. The *capacity region* of a network is defined as the union of the stability region over all policies. If the stability region of an algorithm OJS-ALG and utility function u is equal to the capacity region, we say that policy (OJS-ALG, u) is throughput-optimal.

In order to investigate the network capacity region in more detail, we first introduce some definitions. We denote by $f_n^{(1)}$ the rate (long-term average traffic flow) of single-transmission packets for user n, by $f_n^{(2)}$ the rate of joint-transmission traffic for user n, and by $f_n^{(3)}$ the rate of user-n traffic sent across the backhaul. Define the vector $\boldsymbol{f} = (f_1^{(1)}, f_1^{(2)}, f_1^{(3)}, \ldots, f_N^{(1)}, f_N^{(2)}, f_N^{(3)})$. Then, for a given arrival rate vector $\boldsymbol{\lambda}$, the set of all $\boldsymbol{\lambda}$-admissible traffic flows can be defined as

$$F_{\boldsymbol{\lambda}} = \left\{ \boldsymbol{f} \in \mathbb{R}_+^{3N} \;\middle|\; \lambda_n = f_n^{(1)} + f_n^{(3)},\ f_n^{(3)} = f_n^{(2)},\ n \in \mathcal{N} \right\}. \tag{11}$$

That is, a flow is $\boldsymbol{\lambda}$-admissible if for all queues Q_n, $\widehat{Q}_n$, $n = 1 \ldots, N$, the traffic arrival rate is equal to the departure rate.

We now introduce the set of all arrival rate vectors such that at least one $\boldsymbol{\lambda}$-admissible flow can be realized:

$$\Lambda = \left\{ \boldsymbol{\lambda} \in \mathbb{R}_+^{3N} \mid \exists \boldsymbol{f} \in F_{\boldsymbol{\lambda}}\ \exists \boldsymbol{r} \in \text{conv}(R) f_n^{(j)} < r_n^{(j)} \text{ if } f_n^{(j)} > 0,\ n \in \mathcal{N},\ j = 1, 2, 3 \right\},$$

where conv(R) denotes the convex hull of R, the set of all rates across the various links that can be achieved in saturation.

We now show that any $\boldsymbol{\lambda} \in \Lambda$ can be stabilized, and that any $\boldsymbol{\lambda}$ outside of the closure of Λ cannot. For stabilizing $\boldsymbol{\lambda} \in \Lambda$ we use the policy (OJS-OPT, u_Q), where OJS-OPT represents any algorithm that solves OJS exactly, and u_Q is the queue-length based utility function from (4). The following theorem then implies that Λ is indeed the capacity region, and that (OJS-OPT, u_Q) is throughput-optimal. The proof relies on a standard drift argument using a quadratic Lyapunov function [6].

Theorem 5. *Let $\boldsymbol{\lambda} \in \Lambda$, then the queue-length process is stable under policy (OJS-OPT, u_Q). If $\boldsymbol{\lambda} \notin \bar{\Lambda}$, then there exists no policy that stabilizes the network.*

The reason for choosing the queue-length based utility function (4) becomes clear when substituting it into the objective function of OJS.

$$U(\boldsymbol{z}) = \sum_{n=1}^{N} \Big(L_n(t) \mathbb{E}\{\mu_n^{(1)}(t)\} + \widehat{L}_n(t) \mathbb{E}\{\mu_n^{(3)}(t)\} + \big(L_n(t) - \widehat{L}_n(t)\big) \mu_n^{(2)}(t) \Big). \tag{12}$$

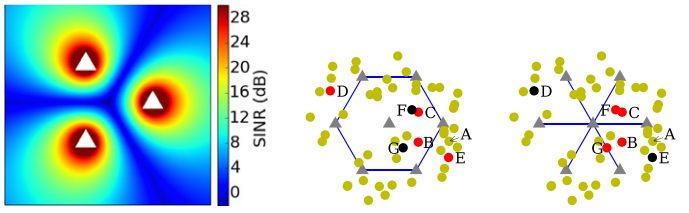
This yields the objective function of the celebrated MaxWeight scheduling algorithm. This algorithm, first introduced in [6], has been shown to be throughput-optimal in a wide range of settings, see, e.g., [6,30]. Note that although our objective function of maximizing the queue-weighted throughput is similar to that used in traditional work on MaxWeight scheduling, the constraints of the OJS problem are markedly different. Specifically, the MaxWeight scheduling literature is typically concerned with maximum (weighted) set problems, which are fundamentally different from the OJS problem.

Since the maximization for utility function (4) is a specific instance of the OJS problem, it follows from Proposition 1 that solving this problem is NP-hard. Thus, using an optimal algorithm OJS-OPT (as in Theorem 5) is typically not feasible in practice for general graphs. In Section 7 we investigate the performance of a wider set of (suboptimal) algorithms via simulation. Theoretical results on the capacity loss for general algorithms will be the subject of future work.

7. Numerical results

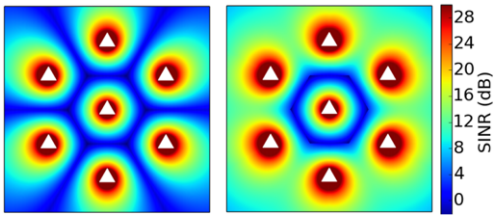
We conducted a simulation study to evaluate the performance of the various algorithms introduced in Section 5. Throughout this section we consider the case where a packet can be transmitted using one of several Modulation and Coding Schemes (MCSs); Details on extending the algorithms to support several MCSs are given in [22]. The simulation results provide insights on the network-level benefits and tradeoffs of JT under various network scenarios.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004



(a) 3-cluster. (b) Cycle topology. (c) Star topology.

Fig. 6. Simulated network topologies. (a) SINR for the cluster of 3 BSs. The dark blue area denotes the location of inter-cell users. (b)–(c) Two 7 BS topologies. The color of the nodes illustrates the throughput gains of users for a sample run. Red users benefit from JT and the throughput of yellow and black users remains roughly the same. Black users have throughput of 0 both with and without JT. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 7. SINR for the cycle topology for (a) single-transmissions and (b) joint-transmissions.

7.1. Simulation setup

OJS algorithm. We implemented the four algorithms presented in Table 1. The majority of the algorithms and the queueing dynamics are implemented in Python, while the A_{MMK} procedure is written in C. We did not implement any JTC algorithms, since it was proved in Section 5 that JTK guarantees that a feasible solution always exists. Furthermore, as we show in Section 7.2.1, the greedy algorithm shown in Table 2 performs well in most considered scenarios.

Network setup. We consider three network topologies. In Section 7.2.2, we analyze a network of 3 BSs, with backhaul links between each pair (Fig. 6(a)). In Section 7.2.3, we look at two different backhaul topologies for a 7 BS network, shown in Fig. 6(b)–(c). The backhaul capacity (BC) is the same for all links.

We use a fixed packet size of 73 bytes, and the backhaul capacities are normalized to units of packets/subframe. The distance between neighboring BSs is 700 m. The height of each BS's antenna is 20 m. The BSs' transmit power is 39 dBm and 30 dBm for the 3 and 7 BSs network, respectively. Lower transmission power is used for the larger network, since more BSs transmit interfering signals.

We simulate $N = 20$ users for the 3-cluster, and $N = 50$ for the 7-cluster topologies. The users are placed uniformly within a circle that contains the entire simulation area, and with radius 1050 m.

Wireless model. We set $S = 50$ scheduled blocks, corresponding to a 10 MHz LTE system. Once the location for a user is determined, the received power level from each BS is computed based on the Hata propagation model [33] which was shown suitable for LTE in urban areas [34]. The power levels from the different BSs are used to compute the SINR for single and joint transmission to the user. The SINR values for single and joint transmissions for the cycle topology are plotted in Fig. 7. Given the SINR values, the success probability for single and joint transmission is computed for each MCS (QPSK-1/2, QAM64-1/2, and QAM64-3/4) using data taken from [35].

Queueing Dynamics. The queueing dynamics are implemented as in Section 6. Unless otherwise noted, packet arrivals for users follow a binomial distribution with $n = 3$ and $p = 0.5$. In every subframe, an algorithm for solving OJS is executed with the utility function (4). Throughout the simulation, we track the *normalized throughput* of a user, defined as the fraction of arrived packets that have been successfully transmitted. The average normalized throughput is computed over all users. The simulation duration is 1000 subframes and each data point is obtained by averaging over 1000 runs.

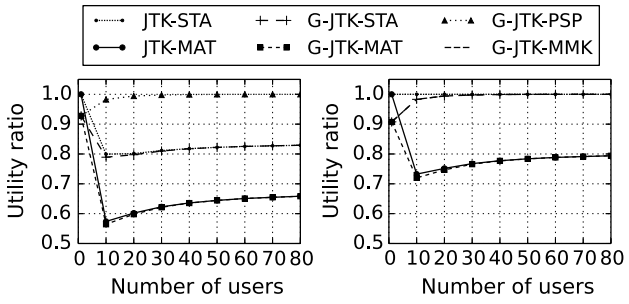
We also distinguish the performance of *inter*-cell users from that of the *intra*-cell users in order to evaluate the benefits of CoMP JT for both types. In this section, we refer to inter-cell users as those whose power levels from two BSs is above a threshold. This threshold is determined numerically by observing the physical location of these users (illustrated by the darker regions of Fig. 6(a)).

7.2. Simulation results

7.2.1. Performance of the approximation algorithms

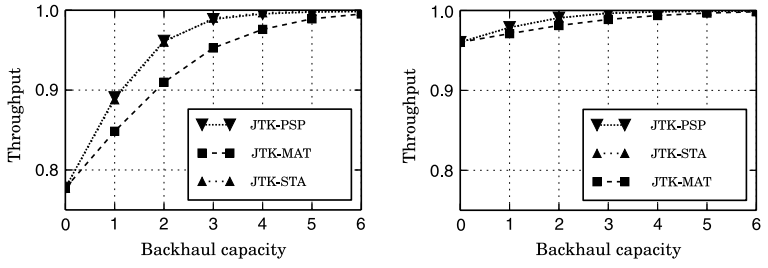
In practice, approximation algorithms may perform significantly better than their guaranteed approximation ratios. Hence, we carry out a single-subframe evaluation with the goal of isolating the performance of the algorithms from the

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004



(a) Complete graph. (b) Bipartite graph.

Fig. 8. The ratio between the optimal utility for the OJS problem for a single subframe, and the utility obtained by the different algorithms under the two topologies.



(a) Inter-cell. (b) Intra-cell.

Fig. 9. The throughput of both intercell and intracell users for a range of backhaul capacity levels.

long-term effects of the queueing dynamics. We consider the OJS formulation with the throughput utility function (3). We consider all algorithms, for two different topologies with 3 BSs: complete graph and a bipartite graph.

When an optimal A_{MMK} subroutine is used, JTK-MMK and JTK-PSP are optimal under the complete graph and bipartite topologies, respectively, by Theorems 1 and 2. These two algorithms, are therefore, used as benchmarks. JTK-MAT and JTK-STA operate on general networks, but only consider a subset of the backhaul links. Therefore, they achieve a fraction of the optimal utility, denoted *utility ratio* in Fig. 8. We vary the number of users from 1 to 80, reflecting the range of users that can be expected in small cell deployments. In each run, a set of items $\mathcal{U}$ is sampled randomly. To obtain a single point, 10,000 iterations are averaged.

We first use the optimal DP algorithm for A_{MMK}, for which $\alpha = 1$ (Table 2). Since the maximum vertex degree is $\Delta(G_J) = 2$ for both topologies under consideration, JTK-MAT and JTK-STA are 1/3- and 1/2-approximations, respectively (Table 1). For the complete graph topology, JTK-MAT achieves a utility ratio of 0.6 at its worst, while JTK-STA does better with a ratio of 0.8 (Fig. 8(a)). Similar insights hold for the bipartite topology in Fig. 8(b).

In addition to using the optimal A_{MMK}, we ran the same simulations for the case when a greedy algorithm is used for A_{MMK} (Table 2), denoted with the prefix "G-" in Fig. 8. In this case the approximation ratios no longer hold, as there are no performance guarantees for the greedy algorithm. However, we found that when A_{MMK} is solved greedily, the algorithms are very close to optimal for more than 10 users. Moreover, the running time of the greedy algorithm is significantly lower than the duration of a subframe. Due to their improved running time, we only use the greedy version in the following sections. For clarity, we omit the "G-" prefix.

7.2.2. *Impact of backhaul capacity*

Backhaul links are typically expensive to deploy, and operators frequently have to lease them. Therefore, it is important for the operator to strike a balance between improving performance and containing backhaul costs. To obtain a better understanding of the required backhaul capacity, we evaluated its impact on the long-term throughput of the queueing system.

In Fig. 9, the user-averaged normalized throughput is shown when the backhaul capacity between each pair of BSs is scaled from 0 to 6 packets/subframe. Inter-cell users in particular gain from JT (Fig. 9(a)) when network-level behavior is considered. A 28% throughput gain is observed for those users with the addition of backhaul. Half of this gain is achievable with 1 unit of backhaul capacity, while 2 units realizes 80% of the potential gains. However, intra-cell users gain 5% throughput. As cell sizes become smaller, the portion of inter-cell users increases and the overall gain from using CoMP JT will be higher.

The achieved throughput depends on the used algorithms. The largest benefits are possible with the JTK-PSP and JTK-STA algorithms, which utilize 3 or 2 backhaul links in every subframe, respectively. It is also observed that in clusters of

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

this size, JTK-STA performs as well as the optimal JTK-PSP, despite its lower running time. JTK-MAT requires more backhaul capacity to achieve the same throughput. Overall, in this case JTK-STA is the best choice while JTK-MAT can be used to reduce computational resources at the cost of a larger investment in infrastructure.

7.2.3. Impact of topologies

We now consider the *star* and *cycle* topologies with 7 BSs, illustrated in Fig. 6(b) and (c), respectively. Since both of these topologies are bipartite, we use the optimal algorithm JTK-MMK in this section.

To study the impact of network topology on JT, we study the throughput gains that are obtained when backhaul links with a 3 unit capacity are introduced in the different topologies. We invoke the algorithm for the same user placement and average the results. The throughput gains of individual users under the cycle and star topologies are illustrated in Fig. 6(b) and (c), respectively. Overall, we see that in each topology there are 4 users that observed an increased throughput from JT while the throughput of the other users remained very similar. While we do not list here the change in throughput for each individual user, the trend is as follows. The red users have throughput of 0 without JT, since the SINR for a single transmission to these users leads to a packet transmission success probability of zero. With JT, the throughput of the red users is very close to 1 and the throughput of each yellow users is reduced by at most 0.001.

In both topologies we see improvement in the throughput of 4 users alongside a negligible decrease in throughput of some of other users. However, we observe that the users that benefit from JT (i.e., red users) are different for the different topologies. The throughput of users B, C, D, and E increased in the cycle topology but for the star topology the throughput of users B, C, F, and G increased. Clearly, this is to be expected as the users who gain from JT are those that reside between two BSs that are connected via a backhaul link. Additionally, we notice that in each topology there are two users that exhibit a throughput of 0 both with and without using JT. For the cycle topology (Fig. 6(b)) the users are F and G, while for the star topology (Fig. 6(c)) the users are D and E. To conclude, the two different topologies result with different distribution of the throughput among the users that are located between the BSs. Since the users are located somewhat evenly in the simulation area, the number of red users is similar for both topologies. However, for a different user placement a specific topology may benefit a larger number of users.

The impact of the topology on the stability region of user A from Fig. 6(b)–(c) is illustrated in Fig. 10(a). The aggregate queue size at the end of the simulation run, under different arrival rates and backhaul capacities, highlights that this user's queues can be stabilized for higher arrival rates under the star topology. This behavior is representative of other inter-cell users.

To further study the performance of our algorithms under the different topologies we plot in Fig. 10(b) and Fig. 10(c) the throughput as a function of the arrival rate, for the cycle and star topologies, respectively. In both figures JT improves the network throughput by 9%, even for low loads. This can be explained by the observation that certain inter-cell users may never receive a packet through single-transmission. For the cycle topology (Fig. 10(b)), the performance of JTK-STA and JTK-MAT is comparable, despite the different approximation ratio. This is because in the cycle topology a matching may include 5 out of the 6 available backhaul links, making JTK-MAT comparable to JTK-STA. It turns out that for higher rate values, JTK-MAT performs slightly better than JTK-STA. Therefore, an operator may choose to run both JTK-MAT and JTK-STA and select the schedule that yields the highest throughput in each subframe. Studying such algorithm is out of scope and left for future research. In the star topology (Fig. 10(c)), since JTK-STA is an optimal algorithm, we omit the curve for JTK-MMK from the figure. For lower arrival rate the performance of JTK-MAT is very close to optimal; As the arrival rate increases JTK-STA is clearly favorable. This is expected since for the star topology the matching computed by JTK-MAT will include at most one backhaul link.

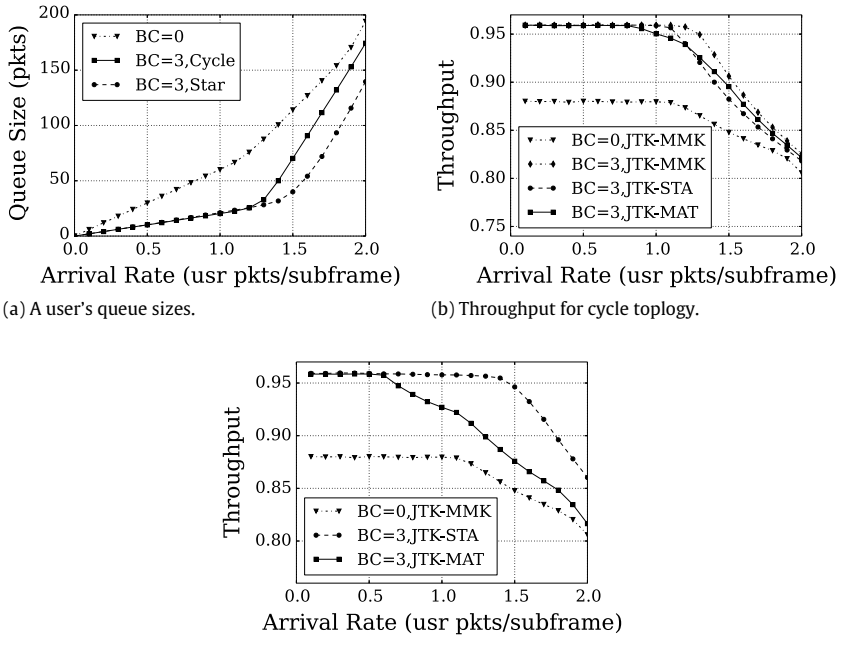
The joint impact of backhaul capacity for star and cycle topologies is shown in Fig. 11(a) and (b), respectively. Recall that JTK-STA is optimal in Fig. 11(b) and note that, compared to the optimal JTK-MMK in Fig. 11(a), backhaul capacity of 3 or more is beneficial only for the cycle topology. This is due to the fact that users residing between the center BS and another BS exhibit greater interference than users that reside between two non-center BSs. Therefore, in the star topology, such users gain significant increase in their throughput by just forwarding their packets. As we observed in Fig. 6(c), there are two such users and since the arrival rate is 1.5, a backhaul capacity of 3 suffices for the star topology. However, in the cycle topology, while mainly two users gain a significant throughput increase, other users which can get their packets using single transmission can still benefit from JT. Therefore, increasing the backhaul capacity further improves the overall throughput.

Fig. 11(a) and (b) also demonstrate the relationship between the network topology and the performance of our algorithms. For the cycle topology (Fig. 11(a)), a matching may include 5 out of the 6 backhaul links, for this reason JTK-MAT is relatively close to optimal and even surpasses JTK-STA for most backhaul capacity values. On the other hand, for star topology (Fig. 11(b)), while JTK-STA is guaranteed to be optimal, the matching computed in JTK-MAT contains at most one backhaul link and therefore the algorithm obtains low performance. Note that JTK-MAT with backhaul capacity 6 performs closely to JTK-STA with backhaul capacity 1 since it utilizes at most one out of the 6 backhaul links in every subframe.

8. Conclusions

In this paper, we considered a cellular network with Coordinated Multi-Point (CoMP) Joint Transmission (JT) capabilities that allow multiple BSs to transmit simultaneously to each user. We first formulated the OFDMA Joint Scheduling (OJS) problem of determining a subframe schedule and deciding if to use JT. By exploiting the characteristics of this problem, we developed efficient scheduling algorithms for bipartite graphs and approximation algorithms for general graphs.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

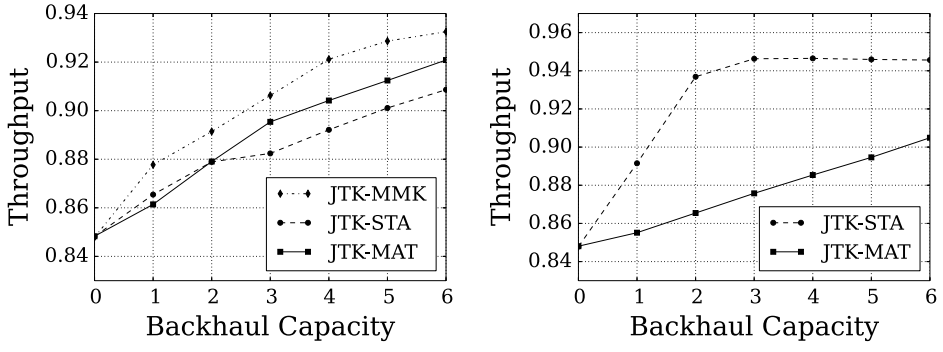


(a) A user's queue sizes.

(b) Throughput for cycle toplogy.

(c) Throughput for star topology.

Fig. 10. (a) The final sum of queue sizes of user A from Fig. 6(b), as the arrival rate is varied. Larger backhaul capacities (BCs) keep the queues bounded for higher arrival rates; (b)–(c) Throughput obtained by the proposed scheduling algorithms vs. the arrival rate for the star and cycle topologies.



(a) Throughput for cycle toplogy.

(b) Throughput for star topology.

Fig. 11. Throughput obtained by the proposed scheduling algorithms vs. the backhaul capacity for the star and cycle topologies, for an arrival rate of 1.5 packets per user per subframe.

We then considered a queueing model that evolves over time. In this model, we proved that solving the OJS problem with a specific queue-based utility function (in every subframe) achieves maximum throughput in CoMP-enabled networks.

Via extensive simulations we showed that the bulk of the gains from CoMP with JT can be achieved with low capacity backhaul links. We showed that our algorithms distribute the network resources evenly, increasing the inter-cell users' throughput at only a slight cost to the intra-cell users.

This paper is the first step towards a rigorous, *network-level* understanding of the impact of cross-layer scheduling algorithms on CoMP networks with JT. In future research, we will extend the model by allowing more than two BSs to jointly transmit, and by allowing longer backhaul delays. Moreover, we will apply our techniques to CoMP related technologies such as network-MIMO, multi-cell MIMO, and MU-MIMO. Finally, we will study the design considerations of the backhaul network and the impact of decentralization on the performance.

Acknowledgments

This work was supported in part by NSF grant CNS-10-54856, CIAN NSF ERC under grant EEC-0812072, and the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme (FP7/20072013) under REA grant agreement no. [PIIF-GA-2013-629740].11. The authors also gratefully acknowledge Marc Kurtz for his contributions to the development of the simulation code and performance evaluation methodology.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

Appendix. Proofs

Proof of Proposition 1. To prove Proposition 1, we use the following definition:

Definition 3. The *chromatic index of a graph G* [24], $\chi'(G)$, is the number of colors required to color the edges of G such that no two adjacent edges have the same color.

It is known by Vizing's theorem that for every simple graph G, $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$, where $\Delta(G)$ is the maximum vertex degree of G. The Minimum Edge Coloring Problem (MECP) [24] is to determine whether $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G)+1$. It is well-known that MECP is NP-hard [24], therefore to complete the proof we present a polynomial-time reduction from MECP to OJS.

Given a simple graph $G = (V, E)$ with maximum vertex degree $\Delta(G)$, we now describe how to construct an OJS instance. We set $\mathcal{B} = V$. For each edge $\{v_1, v_2\} \in E$, we add a user n to $\mathcal{N}$ for which $\text{BS}(n) = v_1$, $\widehat{\text{BS}}(n) = v_2$, and there exists only a single pending packet in $\widehat{Q}_n$. Thus, for every packet i we have $\beta(i) = 1$. The utility is defined as $u(i, 1) = 1$ and $u(i, 0) = 1$ for every i. Also, we set $S = \Delta(G)$, $K = S$, and $\mathcal{C} = \{\{a, b\},\ a, b \in \mathcal{B} \wedge a \neq b\}$. Note that for the constructed instance (a)–(b) stated in the lemma hold.

We now show that the optimal solution to the OJS instance has a utility of $|E|$ if and only if $\chi'(G) = \Delta(G)$. Let $\boldsymbol{x}^*,\ \boldsymbol{y}^*,\ \boldsymbol{z}^*$ be an optimal solution to the OJS instance with a total utility of $|E|$. Due to the utility u used and since the total number of packets is $|E|$, $z_i^* = 1$, $\forall i$. Consider an edge $e = \{v_1, v_2\} \in E$ and a pending packet i' such that $h(i') = \{v_1, v_2\}$. Due to (7), there is exactly one s' for which $x_{i's'} = 1$. We assign e the color s' and continue the process for the remaining edges and packets. Since $1 \leq s' \leq S = \Delta(G)$, at most $\Delta(G)$ colors are used. Since (8) also holds, no two adjacent edges are colored using the same color s. We showed an edge coloring with at most $\Delta(G)$ colors, $\chi'(G) = \Delta(G)$. The other direction, namely showing that if $\chi'(G) = \Delta(G)$ then the optimal solution to the OJS instance has a utility of $|E|$, can be proved similarly. □

Proof of Lemma 1. The proof immediately follows from the definitions of the JTK and JTC problems. □

Proof of Lemma 2. Given a solution $\boldsymbol{x}'$ to a JTC instance, we now define a coloring on G_{SB} that uses at most S colors. Observe that by Definition 2 and since constraint (7) holds, there exists a one-to-one mapping from every pair (i, s) such that $x'_{is} = 1$ into an edge in E_{SB}. This mapping defines an edge coloring using at most S colors ($1 \leq s \leq S$). Since constraint (8) holds, no two edges of the same color touch a vertex in G_{SB}. The other direction, namely, finding a solution $\boldsymbol{x}'$ to JTC, given an edge coloring on G_{SB}, can be proved similarly. □

Proof of Lemma 3. Using Definition 2, it is clear that the subgraph $G' = (V', E')$ of G_{SB} defined by $V' = \mathcal{B}$ and $E' = \{\{a, b\} \in E_{SB} : a, b \in \mathcal{B}\}$ is bipartite. Since each $b \in V_{\text{SB}} \setminus V'$ has at most one neighbor, G_{SB} is bipartite. □

Proof of Lemma 4. Using the result from [25] and since G_{SB} is bipartite and $\Delta(G_{\text{SB}}) \leq S$, G_{SB} has an edge coloring that uses at most S colors. By Lemma 2, such a coloring defines a solution $\boldsymbol{x}'$ such that (7), (8) hold. □

Proof of Theorem 1. We already showed that for bipartite networks JTK-MMK solves JTK and JTC-BIP solves JTC. Lemma 1 concludes the proof. □

Proof of Lemma 5. Let $\boldsymbol{z}^*, \boldsymbol{y}^*$ be an optimal solution for JTK. Without (7) and (8), JTK would be equivalent to MMK with some restrictions on its parameters (unit-size MMK items and two MMK capacity values). Therefore, let $\boldsymbol{z}', \boldsymbol{y}'$ be the solution returned by JTK-MMK, $U(\boldsymbol{z}', \boldsymbol{y}') \geq \alpha U(\boldsymbol{z}^*, \boldsymbol{y}^*)$. Finally, the solution is feasible due to Lemmas 3 and 4. □

Proof of Theorem 3. Let $\mathcal{B}_0$ be the set of BSs with no backhaul links (BSs whose degree is 0 in G_J). In line 2 of JTK-MAT the selected transmissions are determined using A_{MMK}. This set of transmission is an α-approximation with respect to a JTK instance with $\mathcal{B}'' = \mathcal{B}_0$. Therefore, to complete the proof we can assume that every BS has a backhaul link and show that the transmissions determined in line 7 are an $(2\alpha/3\Delta(G_J))$-approximation.

Let $\boldsymbol{z}^*_{\text{OJS}}, \boldsymbol{y}^*_{\text{OJS}}$ be an optimal solution for OJS. The sum of weights for all edges in $\mathcal{C}$, as computed in line 5 of JTK-MAT, is at least $\alpha U(\boldsymbol{z}^*_{\text{OJS}}, \boldsymbol{y}^*_{\text{OJS}})$.

Any graph G has an edge coloring using at most $(\frac{3}{2})\Delta(G)$ colors [36]. Such a coloring for G_J partitions $\mathcal{C}$ into $(\frac{3}{2})\Delta(G_J)$ matchings. Since in line 6 of JTK-MAT a maximum weight matching $\mathcal{E}$ is obtained, the sum of weight for edges in $\mathcal{E}$ is at least $\frac{\alpha U(\boldsymbol{z}^*_{\text{OJS}}, \boldsymbol{y}^*_{\text{OJS}})}{(3/2)\Delta(G_J)} = (2\alpha)/(3\Delta(G_J))U(\boldsymbol{z}^*_{\text{OJS}}, \boldsymbol{y}^*_{\text{OJS}})$. Let $\boldsymbol{z}', \boldsymbol{y}'$ be the solution returned by JTK-MAT. Then, $U(\boldsymbol{z}', \boldsymbol{y}') \geq (2\alpha)/(3\Delta(G_J))U(\boldsymbol{z}^*_{\text{OJS}}, \boldsymbol{y}^*_{\text{OJS}})$.

We now show that $\boldsymbol{z}', \boldsymbol{y}'$ is feasible. Note that (5) and (6) hold since the solution associated with each edge $\{a, b\}$ is feasible (line 5) and solutions of different edges in $\mathcal{E}$ use items whose $h(i)$ is in disjoint (no two edges in $\mathcal{E}$ share a vertex). Due to line 7 of Algorithm JTK-MAT, in the returned solution if $z'_i = 1$ and $h(i) = \{a, b\}$ then $\{a, b\} \in \mathcal{E}$. Since $\mathcal{E}$ is a matching, G_{SB} is bipartite and $\Delta(G_{\text{SB}}) \leq S$. Using Lemma 4 we conclude that $\boldsymbol{z}', \boldsymbol{y}'$ is feasible.

Finally, by Lemma 2 JTC-BIP solves JTC. By applying Lemma 1, we complete the proof. □

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

Proof of Theorem 4. Let $\boldsymbol{z}^*_{\text{OJS}}$ be the optimal solution for OJS. The sum of weights for all vertices in $\mathcal{B}$, as computed in line 5 of JTK-STA, is at least $\alpha U(\boldsymbol{z}^*_{\text{OJS}})$. In each iteration of the repeat loop (line 7 of JTK-STA), in line 10 the utility added to the solution $\boldsymbol{z}''$ equals $U(b_{\max})$ (note that z''_i gets updated to 1 at most once due to the update of I'' in line 12). In line 12, $b_{\max}$ and its neighbors are removed from consideration. The utility lost due to this removal is at most $\Delta(G_J)U(b_{\max})$. Therefore, the total utility of $\boldsymbol{z}''$ returned in line 18 is at least $(\alpha/\Delta(G_J))U(\boldsymbol{z}^*_{\text{OJS}})$.

To show that $\boldsymbol{z}''$ returned by JTK-STA is feasible, it is sufficient to note that due to the correctness of A_{MMK} function SOL-STAR returns a feasible instance with respect to I'' and that $\tilde{\mathcal{I}}$ (line 9) in different iterations contains packets of disjoint sets of transmitting BSs (therefore having positive weight only in disjoint sets of dimensions).

Finally, by Lemma 2 JTC-BIP solves JTC. By applying Lemma 1, we complete the proof. □

Proof of Lemma 6. It is clear that if G_J is planar, G_{SB} is also planar. Therefore, to complete the proof it is sufficient to show that G_{SB} is series–parallel.

To show G_{SB} is series–parallel, we use the following definition [37]. A multigraph is *series–parallel* if it has no subgraph isomorphic to a subdivision of a clique of size 4. Since G_J has no subgraph isomorphic to a subdivision of a clique of size 4, by adding parallel edges to G_J such a subgraph cannot be created in G_{SB}. Therefore, by the above definition G_{SB} is series–parallel. □

Proof of Theorem 2. The following result, mentioned in [28], is needed.

> Let $E_U \subseteq E$ denote edges in E whose both vertices are in U and let $\delta(G) = \max\{\frac{2|E_U|}{|U|-1} : U \subseteq V,\ |U| \geq 3 \text{ and odd}\}$. If G is planar and series–parallel then $\chi'(G) = \max\{\Delta(G), \lceil\delta(G)\rceil\}$ and JTC-PSP from [28] finds an edge coloring that uses $\chi'(G)$ colors.

Recall that in JTK-PSP an instance for MMK with $D + |\mathcal{B}_{\text{odd}}|$ dimensions is constructed. The weight constraints for the new $|\mathcal{B}_{\text{odd}}|$ dimensions are equivalent to requiring that for a feasible solution $\boldsymbol{z}'$, $\delta(G_{\text{SB}}) \leq S$. Therefore, for such G_{SB} there exists an edge coloring that uses at most S colors, and by Lemma 2 such a coloring defines a solution $\boldsymbol{x}'$ such that (7), (8) hold. Since JTK-PSP invokes A_{MMK} which returns an α-approximation solution to the constructed MMK problem, JTK-PSP is an α-approximation for JTK.

Finally, by Lemma 2, JTC-PSP solves JTC. By applying Lemma 1, we complete the proof. □

Proof of Theorem 5. Let $\boldsymbol{\lambda} \in \Lambda$. In order to demonstrate positive recurrence of $\{\boldsymbol{L}(t)\}_{t\geq 0}$, we define a Lyapunov function, and show that it has negative drift outside some closed set of states. Let $\boldsymbol{l} = (l_1, \hat{l}_1, \ldots, l_N, \hat{l}_N)$ and define the quadratic Lyapunov function $V(\boldsymbol{l}) = \sum_{n=1}^{N} l_n^2 + \hat{l}_n^2$.

Consider the one-slot drift

$$\Delta V(\boldsymbol{l}) = \mathbb{E}\{V(\boldsymbol{L}(t+1)) - V(\boldsymbol{L}(t)) \mid \boldsymbol{L}(t) = \boldsymbol{l}\}. \tag{13}$$

By (1) and (2) we compute

$$L_n(t+1)^2 = L_n(t)^2 + 2L_n(t)\Big(W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t)\Big) + \Big(W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t)\Big)^2,$$

$$\widehat{L}_n(t+1)^2 = \widehat{L}_n(t)^2 + 2\widehat{L}_n(t)\Big(\mu_n^{(3)}(t) - \mu_n^{(2)}(t)\Big) + \Big(\mu_n^{(3)}(t) - \mu_n^{(2)}(t)\Big)^2.$$

Substituting this into (13) we obtain

$$\begin{aligned}\Delta V(\boldsymbol{l}) = &\sum_{n=1}^{N} \mathbb{E}\left\{\Big(W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t)\Big)^2 + \Big(\mu_n^{(3)}(t) - \mu_n^{(2)}(t)\Big)^2 \mid \boldsymbol{L}(t) = \boldsymbol{l}\right\} \\ &+ 2\sum_{n=1}^{N} \mathbb{E}\left\{L_n(t)\Big(W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t)\Big) + \widehat{L}_n(t)\Big(\mu_n^{(3)}(t) - \mu_n^{(2)}(t)\Big) \mid \boldsymbol{L}(t) = \boldsymbol{l}\right\}.\end{aligned} \tag{14}$$

Since the $W_n(t)$ have finite second moment, and the $\mu_n^{(j)}(t)$ have finite support, we can bound (for some constant $C < \infty$),

$$\sum_{n=1}^{N} \mathbb{E}\left\{\Big(W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t)\Big)^2 + \Big(\mu_n^{(3)}(t) - \mu_n^{(2)}(t)\Big)^2 \mid \boldsymbol{L}(t) = \boldsymbol{l}\right\} < C.$$

The second part of (14) can be written as

$$\begin{aligned}&2\sum_{n=1}^{N} \mathbb{E}\left\{L_n(t)\Big(W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t)\Big) + \widehat{L}_n(t)\Big(\mu_n^{(3)}(t) - \mu_n^{(2)}(t)\Big) \mid \boldsymbol{L}(t) = \boldsymbol{l}\right\} \\ &= 2\sum_{n=1}^{N}\Big(l_n\lambda_n - l_n\mathbb{E}\{\mu_n^{(1)}(t) + \mu_n^{(3)}(t) \mid \boldsymbol{L}(t) = \boldsymbol{l}\} - \hat{l}_n\mathbb{E}\{\mu_n^{(3)}(t) - \mu_n^{(2)}(t) \mid \boldsymbol{L}(t) = \boldsymbol{l}\}\Big).\end{aligned} \tag{15}$$

Since $\boldsymbol{\lambda} \in \Lambda$, we know by the definition of Λ that there exists a flow vector $\boldsymbol{f}$ such that the conditions in (11) hold. Moreover, $\boldsymbol{\lambda} \in \Lambda$ also implies that there exists a $\boldsymbol{r} \in \text{Conv}(R)$ such that $f_n^{(j)} < r_n^{(j)}$ if $f_n^{(j)} > 0$, $j = 1, 2, 3$, $n = 1, \ldots, N$. Since $\boldsymbol{f}$ is dominated by $\boldsymbol{r}$, and $\boldsymbol{r} \in \text{Conv}(R)$, there exist $\sigma_1, \ldots, \sigma_R$ such that

$$\boldsymbol{f} = \sum_{i=1}^{|R|} \sigma_i \boldsymbol{r}_i, \qquad \sum_{i=1}^{|R|} \sigma_i < 1, \tag{16}$$

where $\boldsymbol{r}_i = (r_{i,1}^{(1)}, r_{i,1}^{(2)}, r_{i,1}^{(3)}, \ldots, r_{i,N}^{(1)}, r_{i,N}^{(2)}, r_{i,N}^{(3)})$ represents the ith vector in R.

Using (16) we obtain

$$l_n \lambda_n = l_n (f_n^{(1)} + f_n^{(3)}) = l_n \sum_{i=1}^{|R|} \sigma_i \left(r_{i,n}^{(1)} + r_{i,n}^{(3)}\right), \tag{17}$$

$$0 = \hat{l}_n (f_n^{(2)} - f_n^{(3)}) = \hat{l}_n \sum_{i=1}^{|R|} \sigma_i \left(r_{i,n}^{(2)} - r_{i,n}^{(3)}\right). \tag{18}$$

By combining (17) and (18), and exploiting the structure of the $\boldsymbol{\mu}$ chosen according to u_Q (see (12))

$$\begin{aligned} 2 \sum_{n=1}^{N} l_n \lambda_n &= 2 \sum_{n=1}^{N} \sum_{i=1}^{|R|} \sigma_i \left(l_n (r_{i,n}^{(1)} + r_{i,n}^{(3)}) + \hat{l}_n (r_{i,n}^{(2)} - r_{i,n}^{(3)}) \right) \\ &\leq 2 \sum_{n=1}^{N} \left(l_n (\mathbb{E}\{\mu_n^{(1)}(t) + \mu_n^{(3)}(t) \mid \boldsymbol{L}(t) = \boldsymbol{l}\}) + \hat{l}_n (\mathbb{E}\{\mu_n^{(2)}(t) - \mu_n^{(3)}(t) \mid \boldsymbol{L}(t) = \boldsymbol{l}\}) \right) \sum_{i=1}^{|R|} \sigma_i. \end{aligned} \tag{19}$$

Substituting (19) into (15) yields

$$\begin{aligned} &2 \sum_{n=1}^{N} \mathbb{E} \left\{ L_n(t) \left(W_n(t) - \mu_n^{(1)}(t) - \mu_n^{(3)}(t) \right) + \widehat{L}_n(t) \left(\mu_n^{(3)}(t) - \mu_n^{(2)}(t) \right) \mid \boldsymbol{L}(t) = \boldsymbol{l} \right\} \\ &\leq -2 \left(1 - \sum_{i=1}^{|R|} \sigma_i \right) \sum_{n=1}^{N} \left(l_n \mathbb{E}\{\mu_n^{(1)}(t) + \mu_n^{(3)}(t) \mid \boldsymbol{L}(t) = \boldsymbol{l}\} - \hat{l}_n \mathbb{E}\{\mu_n^{(3)}(t) - \mu_n^{(2)}(t) \mid \boldsymbol{L}(t) = \boldsymbol{l}\} \right) < 0, \end{aligned} \tag{20}$$

where the last inequality follows from the choice of $\mu_n^{(j)}$ (12). The expression in (20) can be made arbitrarily small by increasing the state $\boldsymbol{l}$. Positive recurrence of $\{\boldsymbol{L}(t)\}_{t \geq 0}$ then follows from [38, Theorem 2.2.4]. □

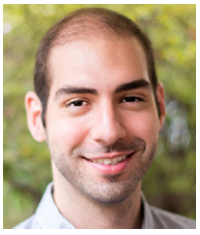
References

[1] 3GPP. Technical Specification Group Radio Access Network; Coordinated multi-point operation for LTE physical layer aspects (Release 11), TR 36.819, September 2013.
[2] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, V. Jungnickel, Coordinated multipoint: Concepts, performance, and field trial results, IEEE Commun. Mag. 49 (2) (2011) 102–111.
[3] K. Kwak, H. Lee, H.W. Je, J. Hong, S. Choi, Adaptive and distributed CoMP scheduling in LTE-Advanced systems, in: Proc. IEEE VTC'13, June 2013.
[4] S. Fu, B. Wu, H. Wen, P. Ho, G. Feng, Transmission scheduling and game theoretical power allocation for interference coordination in CoMP, IEEE Trans. Wirel. Commun. 13 (1) (2014) 112–123.
[5] F. Zhuang, V.K. Lau, Backhaul limited partial cooperations for MIMO cellular networks via semidefinite relaxation, IEEE Trans. Signal Process. 62 (3) (2014) 684–693.
[6] L. Tassiulas, A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, IEEE Trans. Automat. Control 37 (12) (1992) 1936–1948.
[7] O. Tipmongkolsilp, S. Zaghloul, A. Jukan, The evolution of cellular backhaul technologies: current issues and future trends, IEEE Commun. Surv. Tutor. 13 (1) (2011) 97–113.
[8] Q. Cui, S. Yang, Y. Xu, X. Tao, B. Liu, An effective inter-cell interference coordination scheme for downlink CoMP in LTE-A systems, in: Proc. IEEE VTC'11, May 2011.
[9] Y.-P. Zhang, L. Xia, P. Zhang, S. Feng, J. Sun, X. Ren, Joint transmission for LTE-advanced systems with non-full buffer traffic, in: Proc. IEEE VTC'12, September 2012.
[10] J. Li, T. Svensson, C. Botella, T. Eriksson, X. Xu, X. Chen, Joint scheduling and power control in coordinated multi-point clusters, in: Proc. IEEE VTC'11, May 2011.
[11] J. Yu, Q. Zhang, P. Chen, B. Cao, Y. Zhang, Dynamic joint transmission for downlink scheduling scheme in clustered CoMP cellular, in: Proc. IEEE ICCC'13, August 2013.
[12] S. Brueck, L. Zhao, J. Giese, M.A. Amin, Centralized scheduling for joint transmission coordinated multi-point in LTE-advanced, in: Proc. IEEE WSA'10, February 2010.
[13] R. Cohen, G. Grebla, Joint scheduling and fast cell selection in OFDMA wireless networks, IEEE/ACM Trans. Netw. 23 (1) (2015) 114–125.
[14] R. Cohen, G. Grebla, Multi-dimensional OFDMA scheduling in a wireless network with relay nodes, in: Proc. IEEE INFOCOM'14, April 2014, pp. 2427–2435.
[15] M. Andrews, L. Zhang, Scheduling algorithms for multi-carrier wireless data systems, in: Proc. ACM MOBICOM'07, September 2007.
[16] H.V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, G. Caire, Achieving high data rates in a distributed MIMO system, in: Proc. ACM MOBICOM'12, 2012.
[17] X. Zhang, K. Sundaresan, M.A. Khojastepour, S. Rangarajan, K.G. Shin, NEMOx: scalable network MIMO for wireless networks, in: Proc. ACM MOBICOM'13, September 2013.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004

[18] D. Gesbert, S. Hanly, H. Huang, S. Shamai, O. Simeone, W. Yu, Multi-cell MIMO cooperative networks: A new look at interference, IEEE J. Sel. Areas Commun. 28 (9) (2010) 1380–1408.
[19] M. Di Renzo, H. Haas, A. Ghrayeb, S. Sugiura, L. Hanzo, Spatial modulation for generalized MIMO: challenges, opportunities, and implementation, Proc. IEEE 102 (1) (2014) 56–103.
[20] V. Cadambe, S.A. Jafar, S. Shamai, Interference alignment on the deterministic channel and application to fully connected AWGN interference networks, in: Proc. ITW'08, May 2008.
[21] P. Marsch, G. Fettweis, On multicell cooperative transmission in backhaul-constrained cellular systems, Ann. Telecommun. 63 (5–6) (2008) 253–269.
[22] G. Grebla, B. Birand, P.M. van de Ven, G. Zussman, Joint transmission in cellular networks with CoMP—stability and scheduling algorithms, arXiv:1507.02199, available at http://arxiv.org/abs/1507.02199, 2015.
[23] F. Ren, Y. Xu, H. Yang, J. Zhang, C. Lin, Frequency domain packet scheduling with stability analysis for 3GPP LTE uplink, IEEE Trans. Mob. Comput. 12 (12) (2013) 2412–2426.
[24] I. Holyer, The NP-completeness of edge-coloring, SIAM J. Comput. 10 (1981) 718–720.
[25] R. Cole, K. Ost, S. Schirra, Edge-coloring bipartite multigraphs in $O(E \log D)$ time, Combinatorica 21 (1) (2001) 5–12.
[26] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack Problems, Springer, 2004.
[27] B. Patt-Shamir, D. Rawitz, Vector bin packing with multiple-choice, Discrete Appl. Math. 160 (10–11) (2012) 1591–1600.
[28] X. Zhou, S.-I. Nakano, H. Suzuki, T. Nishizeki, An efficient algorithm for edge-coloring series–parallel multigraphs, in: Proc. LATIN'92, in: LNCS, vol. 583, 1992, pp. 516–529.
[29] Z. Galil, S. Micali, H. Gabow, An $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs, SIAM J. Comput. 15 (1) (1986) 120–130.
[30] N. McKeown, V. Anantharam, J. Walrand, Achieving 100% throughput in an input-queued switch, in: Proc. IEEE INFOCOM'96, March 1996.
[31] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, P. Whiting, Scheduling in a queuing system with asynchronously varying service rates, Probab. Engrg. Inform. Sci. 18 (02) (2004) 191–217.
[32] A. Eryilmaz, R. Srikant, J. Perkins, Stable scheduling policies for fading wireless channels, IEEE/ACM Trans. Netw. 13 (2) (2005) 411–424.
[33] M. Hata, Empirical formula for propagation loss in land mobile radio services, IEEE Trans. Veh. Technol. 29 (3) (1980) 317–325.
[34] N. Tabia, A. Gondran, O. Baala, A. Caminada, Interference model and evaluation in LTE networks, in: Proc. IFIP WMNC'11, October 2011.
[35] K. Balachandran, D. Calin, F.-C. Cheng, N. Joshi, J.H. Kang, A. Kogiantis, K. Rausch, A. Rudrapatna, J.P. Seymour, J. Sun, Design and analysis of an IEEE 802.16e-based OFDMA communication system, BLTJ 11 (4) (2007).
[36] C.E. Shannon, A theorem on coloring the lines of a network, J. Math. Phys. 28 (1949) 148–151.
[37] X. Zhou, Y. Matsuo, T. Nishizeki, List total colorings of series–parallel graphs, J. Discrete Algorithms 3 (1) (2005) 47–60.
[38] G. Fayolle, V. Malyshev, M. Menshikov, Topics in the Constructive Theory of Countable Markov Chains, Cambridge University Press, Cambridge, UK, 1995.

Guy Grebla received the B.A., M.A., and Ph.D. degrees in Computer Science from the Technion—Israel Institute of Technology, completing his Ph.D. studies in 2013. He is currently a postdoctoral research scientist in the Electrical Engineering department at Columbia University, New York, NY. He is an IEEE Senior Member and serves as a Technical Program Committee (TPC) member of IEEE INFOCOM'16 and ACM MOBIHOC'15.

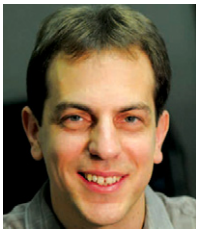


Berk Birand received the Ph.D. degree from the Columbia University Department of Electrical Engineering in May 2015. He obtained a dual B.S. degree in electrical & computer engineering and computer science with distinction from the Worcester Polytechnic Institute in 2008.

He received the Jury Award for his Ph.D. thesis "Cross-layer resource allocation algorithms in wireless and optical networks". He was an IBM Ph.D. Fellow for the 2011–2012 academic year and was a research intern at the IBM T.J. Watson Research Center in 2010 and 2011. He was awarded the "Best Theory Session Talk Award" in the ACM MobiHoc'10 S3 workshop, and the "Millman Award for Outstanding Teaching Assistant" by Columbia University. He was a co-chair of the ACM S3'11 workshop co-located with ACM MobiCom'11.

Peter van de Ven is a tenure-track researcher at CWI, Amsterdam. He received the Ph.D. degree in applied mathematics at Eindhoven University of Technology in 2011, and worked at the IBM T.J. Watson Research Center in NY as a postdoctoral researcher (2011–2013) and research staff member (2013–2014). His research interests are in modeling and analysis of large stochastic systems across different applications.

Dr. Van de Ven is the recipient of the Beta Ph.D. award 2012 for best Ph.D. thesis of the Beta research school, the IBM Goldstine Fellowship Award 2011–2012, and runner-up for the VvS + OR Ph.D. thesis award for best thesis in The Netherlands in OR and statistics, 2010–2012.



Gil Zussman received the Ph.D. degree in electrical engineering from the Technion in 2004 and was a postdoctoral associate at MIT in 2004–2007. He is currently an Associate Professor of Electrical Engineering at Columbia University. He is a co-recipient of 5 paper awards including the ACM SIGMETRICS'06 Best Paper Award and the 2011 IEEE Communications Society Award for Outstanding Paper on New Communication Topics. He received the Fulbright Fellowship, two Marie Curie Fellowships, the DTRA Young Investigator Award, and the NSF CAREER Award, and was a member of a team that won first place in the 2009 Vodafone Foundation Wireless Innovation Project competition.

Please cite this article in press as: G. Grebla, et al., Joint transmission in cellular networks with CoMP—Stability and scheduling algorithms, Performance Evaluation (2015), http://dx.doi.org/10.1016/j.peva.2015.06.004