Joint Selection And Scheduling of Communication Requests in Multi-Channel Wireless Networks under SINR Model

Peng-Jun Wan*†, Huaqiang Yuan*, Jiliang Wang‡, Ju Ren§, and Yaoxue Zhang§
* School of Computer Science, Dongguan University of Technology, P. R. China
† Department of Computer Science, Illinois Institute of Technology
‡ School of Software, Tsinghua University, P. R. China
§ School of Information Science and Engineering, Central South University, P. R. China

***Abstract*—Consider a set of communication requests in a multi-channel wireless network, each of which is associated with a traffic demand of at most one unit of transmission time, and a weight representing the utility if its demand is fully met. A subset of them is said to be *feasible* if they can be scheduled within one unit of time. The problem Maximum-Weighted Feasible Set (MWFS) seeks a feasible subset with maximum total weight together with a transmission schedule of them whose length is at most one unit of time. This paper develops an efficient $\mathcal{O}\left(\log^2 \alpha\right)$-approximation algorithms for the problem MWFS under the physical interference model (aka, SINR model) with a fixed monotone and sub-linear power assignment, where α is the maximum number of requests which can transmit successfully at the same time over the same channel.**

I. Introduction

Consider a set of point-to-point communication requests in a multi-channel wireless network. Each request is associated with a traffic demand of at most one unit of transmission time, and a weight representing the utility if its demand is *fully* met. A subset of them is said to be *feasible* if they can be scheduled within one unit of time. The problem **Maximum-Weighted Feasible Set** (**MWFS**) seeks a feasible subset with maximum total weight together with a transmission schedule of them whose length is at most one unit of time. The problem **MWFS** arises from many fundamental applications in spectrum allocation and wireless scheduling [13], [16]. Technically, it is a non-trivial generalization of the well-known **Maximum-Weighted Independent Set** (**MWIS**) [14], which is a special case of **MWFS** in which all requests have unit traffic demands and transmit over the same channel. On the other hand, the strict harder-ness of **MWFS** than **MWIS** is witnessed by the restriction to simple instances of mutually disjoint but conflicting communication requests in which no pair can transmit at the same time. For any such restricted instance, a maximum weighted independent set is trivially the singleton communication request with the largest weight; the **MWFS** is essentially equivalent to the classic **Knapsack** problem [7], and hence is NP-complete.

Variants of the problem **MWFS** under *protocol* interference model has been recently studied in [13], [16]. To the best of our knowledge, the problem **MWFS** under *physical* interference model (aka, SINR model) has not been studied before. Wireless communication scheduling under physical interference model is notoriously hard due to the non-locality and the additive nature of the wireless interference under the physical interference model. As such, both the design and analysis of the algorithms developed in [13], [16] for **MWFS** under *protocol* interference model cannot be extended directly to the setting of physical interference model. Indeed, the works in [13], [16] heavily rely on two key properties of *protocol* interference model:

- the equivalence of the independence and the inductive independence (to be defined precisely in III-B),
- the availability of the explicit bound of transmission schedule length in terms of the transmission demands.

However, under the physical interference model, there is a significant gap between independence and inductive independence, and no known transmission scheduling algorithm is able to provide an explicit bound on the length of the produced transmission schedule in terms of the transmission demands. The main objective of this paper is to develop new algorithmic techniques for achieving efficient and provably good approximations for the problem **MWFS** under the physical interference model with a fixed monotone and sub-linear power assignment [2], [8], [9], [10]. This specific problem is described as follows.

Suppose that A is a set of communication requests and λ is the number of available channels. Each request a has a demand $d\left(a\right) \in (0,1]$ of transmission time, a positive weight $w\left(a\right)$ of utility, and a transmission power $p\left(a\right)$. The weight of any subset F of A is

$$w\left(F\right) := \sum_{a \in F} w\left(a\right).$$

For each request a, $\ell\left(a\right)$ denotes the distance between the sender and receiver of a. For any two requests a and b in A, $\ell\left(a,b\right)$ denotes the distance between the sender of a and the receiver of b. The signal strength attenuates with a path loss factor $\eta r^{-\kappa}$, where r is the distance from the transmitter, κ is *path-loss exponent* (a constant between 2 and 5 depending on the wireless environment), and η is the *reference loss factor*. The signal quality perceived by a receiver is measured by the *signal to interference and noise ratio* (*SINR*), which is the quotient between the power of the wanted signal and the total power of unwanted signals and the ambient noise ξ. In order to correctly interpret the wanted signal under the physical interference model, the SINR must be greater than certain threshold $\sigma > 1$. Thus, for each request a to communicate even without any interference, $p(a)$ should exceed

$$p_0(a) := \frac{\sigma\xi}{\eta}\ell(a)^{\kappa}.$$

The interference of a request $a \in A$ toward another request $b \in A$ is $p(a) \cdot \eta\ell(a,b)^{-\kappa}$ when they transmit at the same time over the same channel. A set I of requests in A is *independent* if when all requests in I transmit at the same time over the same channel, the SINR of each request in I is greater than σ.

A set C of requests is said to be *compatible* if no pair of requests in C have a primary conflict, and C can be partitioned into λ independent subsets. Physically, compatible requests correspond to those which can transmit successfully over λ channels at the same time. A (transmission) *schedule* of a subset $F \subseteq A$ is a sequence $\mathcal{S}$ of pairs (C_j, x_j) for $1 \leq j \leq k$ where each C_j is a compatible subset of F and x_j is a positive number satisfying that for each $a \in F$,

$$d(a) = \sum_{j=1}^{k} x_j \left|\{a\} \cap C_j\right|;$$

the value $\sum_{j=1}^{k} x_j$ are referred to as the *length* of $\mathcal{S}$, and is denoted by $\|\mathcal{S}\|$. A set of requests F is said to be *feasible* if F admits a schedule of length at most one. Throughout of this paper, the power assignment p is assumed to be *monotone*, i.e. $p(a)$ is non-decreasing with $\ell(a)$, and to be *sub-linear*, i.e., $p(a)\ell(a)^{-\kappa}$ is non-increasing with $\ell(a)$. The problem **MWFS** seeks a feasible subset F of A with maximum weight $w(F)$ together with a schedule of F with length at most one. This paper develops an efficient $\mathcal{O}(\log^2 \alpha)$-approximation algorithm for the problem **MWFS**, where α is the maximum size of the independent subsets of A. Innovative variants of the local-ratio scheme [1], [3], [4], [6], [17], which is equivalent to the primal-dual scheme [5], are utilized in our algorithm design and analyses.

The remainder of this paper is organized as follows. In Section II we give an approximation algorithm for seeking a maximum-weighted compatible subset of requests. Such algorithm also provides an $\mathcal{O}(\log^2 \alpha)$-approximation algorithm for the problem **MWFS** when all requests have demands at least $\Omega(1/\log\alpha)$. In Section III, we present a greedy $\mathcal{O}(\log^2 \alpha)$-approximation algorithm for seeking a shortest schedule of a subset of requests. An advantage of this scheduling algorithm is that the produced schedule admits an explicit upper bound on its length in terms of the demands, which provides an approximate feasibility test. In Section IV, we introduce a notion of constrained inductive feasibility, and develop an approximation algorithm for seeking a maximum-weighted inductively feasible subset of requests. This algorithm is applicable to selection of low-demanded requests. In Section V, we present our main result on joint selection and scheduling algorithm for the problem **MWFS**. It is a special divide-and-conquer $\mathcal{O}(\log^2 \alpha)$-approximation algorithm, and is built upon all the algorithms developed in the preceding sections. We conclude this paper in Section VI.

We conclude this section with the a few notations adopted throughout this paper. Let $\prec$ be an request ordering in the decreasing order of $\ell(a)$. For any pair of requests a and b, both $a \prec b$ and $b \succ a$ represent that a appears before b in the ordering $\prec$. For any $a \in A$ and any $B \subseteq A$, we use $B_{\prec a}$ (respectively, $B_{\succ a}$) to denote the set of $b \in B$ satisfying that $b \prec a$ (respectively, $b \succ a$); in addition, $B_{\preceq a}$ denotes $\{a\} \cup B_{\prec a}$, and $B_{\succeq a}$ denotes $\{a\} \cup B_{\succ a}$. For any positive integer k, $[k]$ denotes the set of positive integers at most k; for any two positive integers k and l with $k \leq l$, $[k,l]$ denotes the set of integers between (including) k and l. Consider a non-empty set E.

- For any real-valued function f on E and any $S \subseteq E$, $f(S)$ represents $\sum_{e \in S} f(S)$ with $f(\emptyset) = 0$.
- For any real-valued function f on $E \times E$, any $a \in E$, and any $S \subseteq E$, $f(S,a)$ represents $\sum_{b \in S} f(b,a)$ with $f(\emptyset, a) = 0$.

II. Maximum-Weighted Compatible Subset

In this section, we develop an approximation algorithm for the problem **Maximum-Weighted Compatible Subset** (**MWCS**): Given a subset of B, seek a compatible subset of B with maximum total weight. This problem is a special case of the problem **MWFS** in which all requests have unit demands, but a multi-channel generalization of the problem **MWIS**. Suppose that there is a β-approximation algorithm $\mathcal{A}$ for the problem **MWIS**. We shall develop a $(\beta + 2)$-approximation algorithm $\mathcal{A}^*$ for the problem **MWCS**.

For each $a \in B$, let $\Gamma(a)$ denote the set of requests in B which have a primary conflict with a; and let $\Gamma[a]$ denote $\Gamma(a) \cap \{a\}$. The algorithm $\mathcal{A}^*$ is based on the local-ratio strategy and proceeds in two phases:

1) **Growing Phase**: This phase produces λ independent sets $S_1, S_2, \cdots, S_\lambda$ sequentially in λ iterations. For each $1 \leq j \leq \lambda$, the j-th iteration first computes a "discounted" weight function w_j; and then produces an independent set S_j by applying the algorithm $\mathcal{A}$ to the instance consisting of

$$A_j := \{a \in A : w_j(a) > 0\}$$

and w_j. The "discounted" weight function w_j is computed as follows:

- $w_1 = w$.
- For each $2 \leq j \leq \lambda$,
$$w_j(a) := w(a) - \sum_{i \in [j-1]} w_i (S_i \cap \Gamma[a]), \forall a \in A.$$

2) **Pruning Phase**: This phase produces λ independent sets $I_\lambda, I_{\lambda-1}, \cdots, I_1$ sequentially satisfying that no pair of requests from these λ sets have a primary conflict; and the union of them, denoted by I, is returned as the output. They are produced as follows:
   - I_λ is simply S_λ.
   - For each integer j from $\lambda - 1$ down to 1, I_j consists of all requests in S_j which have no primary conflict with any request in $\bigcup_{i \in [j+1,\lambda]} I_j$.

Now, we provide a lower bounds on $w(I)$ in terms of the maximum weight of compatible subsets of B.

Theorem 2.1: Let O be a maximum-weighted compatible subset of B. Then,
$$w(I) \geq \frac{w(O)}{\beta + 2}.$$

Proof. We shall prove the theorem by establishing the relations
$$w(I) \geq \textstyle\sum_{j \in [\lambda]} w_j(S_j) \geq \frac{w(O)}{\beta + 2}.$$

We first claim that for any subset C of B and for any partition $C_1, C_2, \cdots, C_\lambda$ of C,
$$\begin{aligned} & w(C) - \textstyle\sum_{j \in [\lambda]} w_j(C_j) \\ & = \textstyle\sum_{j \in [\lambda-1]} \sum_{a \in S_j} \left| \Gamma[a] \cap \left(\bigcup_{i \in [j+1,\lambda]} C_i \right) \right| w_j(a). \end{aligned}$$

Since $w = w_1$, we have
$$\begin{aligned} & w(C) - \textstyle\sum_{j \in [\lambda]} w_j(C_j) \\ & = \textstyle\sum_{j \in [\lambda]} w(C_j) - \sum_{j \in [\lambda]} w_j(C_j) \\ & = \textstyle\sum_{j \in [2,\lambda]} w(C_j) - \sum_{j \in [\lambda-2]} w_j(C_j) \\ & = \textstyle\sum_{j \in [2,\lambda]} [w(C_j) - w_j(C_j)] \\ & = \textstyle\sum_{j \in [2,\lambda]} \sum_{a \in C_j} [w(a) - w_j(a)]. \end{aligned}$$

Using the relations between the original weights and the discounted weights, we have
$$\begin{aligned} & w(C) - \textstyle\sum_{j \in [\lambda]} w_j(C_j) \\ & = \textstyle\sum_{j \in [2,\lambda]} \sum_{a \in C_j} \sum_{i \in [j-1]} w_i(\Gamma[a] \cap S_i) \\ & = \textstyle\sum_{i \in [\lambda-1]} \sum_{j \in [i+1,\lambda]} \sum_{a \in C_j} w_i(\Gamma[a] \cap S_i) \\ & = \textstyle\sum_{j \in [2,\lambda]} \sum_{i \in [j-1]} \sum_{a \in C_j} \sum_{C \in \Gamma[a] \cap S_i} w_i(b). \end{aligned}$$

Note that for any pair of requests a and b,
$$\begin{aligned} & a \in C_j, b \in \Gamma[a] \cap S_i \\ & \Leftrightarrow b \in S_i, a \in C_j \cap \Gamma[b]. \end{aligned}$$

Thus,
$$\begin{aligned} & w(C) - \textstyle\sum_{j \in [\lambda]} w_j(C_j) \\ & = \textstyle\sum_{i \in [\lambda-1]} \sum_{j \in [i+1,\lambda]} \sum_{b \in S_i} \sum_{a \in C_j \cap \Gamma[b]} w_i(b) \\ & = \textstyle\sum_{i \in [\lambda-1]} \sum_{b \in S_i} \sum_{j \in [i+1,\lambda]} \sum_{a \in C_j \cap \Gamma[b]} w_i(b) \\ & = \textstyle\sum_{i \in [\lambda-1]} \sum_{b \in S_i} \left| \Gamma[b] \cap \left(\bigcup_{j \in [i+1,\lambda]} C_j \right) \right| w_i(b) \\ & = \textstyle\sum_{j \in [\lambda-1]} \sum_{a \in S_j} \left| \Gamma[a] \cap \left(\bigcup_{i \in [j+1,\lambda]} C_i \right) \right| w_j(a). \end{aligned}$$

So, the claim holds.

Next, we prove that
$$w(I) \geq \textstyle\sum_{j \in [\lambda]} w_j(S_j).$$

Note that $I_\lambda = S_\lambda$; and for each $j \in [\lambda - 1]$ and any $a \in S_j \setminus I_j$,
$$\left| \Gamma[a] \cap \left(\bigcup_{i \in [j+1,\lambda]} I_i \right) \right| \geq 1.$$

Thus,
$$\begin{aligned} & w(I) - \textstyle\sum_{j \in [\lambda]} w_j(I_j) \\ & = \textstyle\sum_{j \in [\lambda-1]} \sum_{a \in S_j} \left| \Gamma[a] \cap \left(\bigcup_{i \in [j+1,\lambda]} I_i \right) \right| w_j(a) \\ & \geq \textstyle\sum_{j \in [\lambda-1]} \sum_{a \in S_j \setminus I_j} \left| \Gamma[a] \cap \left(\bigcup_{i \in [j+1,\lambda]} I_i \right) \right| w_j(a) \\ & \geq \textstyle\sum_{j \in [\lambda-1]} \sum_{a \in S_j \setminus I_j} w_j(a) \\ & = \textstyle\sum_{j \in [\lambda-1]} w_j(S_j \setminus I_j), \end{aligned}$$

which implies that
$$\begin{aligned} w(I) & \geq \textstyle\sum_{j \in [\lambda]} w_j(I_j) + \sum_{j \in [\lambda-1]} w_j(S_j \setminus I_j) \\ & = w_\lambda(I_\lambda) + \textstyle\sum_{j \in [\lambda-1]} w_j(S_j) \\ & = w_\lambda(S_\lambda) + \textstyle\sum_{j \in [\lambda-1]} w_j(S_j) \\ & = \textstyle\sum_{j \in [\lambda]} w_j(S_j). \end{aligned}$$

Finally, we prove that
$$w(O) \leq (\beta + 2) \textstyle\sum_{j \in [\lambda]} w_j(S_j).$$

Let $\{O_1, O_2, \cdots, O_\lambda\}$ be a partition of O into independent sets. Then, for any $j \in [\lambda]$,
$$w_j(O_j) \leq w_j(O_j \cap A_j) \leq \beta w_j(S_j);$$
and for any $a \in S_j$,
$$\left| \Gamma[a] \cap \left(\bigcup_{i \in [j+1,\lambda]} O_i \right) \right| \leq |\Gamma[a] \cap O| \leq 2.$$

Thus,
$$\begin{aligned} & w(O) - \textstyle\sum_{j \in [\lambda]} w_j(O_j) \\ & = \textstyle\sum_{j \in [\lambda-1]} \sum_{a \in S_j} \left| \Gamma[a] \cap \left(\bigcup_{i \in [j+1,\lambda]} O_i \right) \right| w_j(a) \\ & \leq 2 \textstyle\sum_{j \in [\lambda-1]} \sum_{a \in S_j} w_j(a) \\ & = 2 \textstyle\sum_{j \in [\lambda-1]} w_j(S_j), \end{aligned}$$

which implies that
$$\begin{aligned} w(O) & \leq \textstyle\sum_{j \in [\lambda]} w_j(O_j) + 2 \sum_{j \in [\lambda-1]} w_j(S_j) \\ & = \beta \textstyle\sum_{j \in [\lambda]} w_j(S_j) + 2 \sum_{j \in [\lambda-1]} w_j(S_j) \\ & \leq (\beta + 2) \textstyle\sum_{j \in [\lambda]} w_j(S_j). \end{aligned}$$

This completes the proof the theorem. ■

Next, we provide a lower bounds on $w(I)$ in terms of the maximum weight of feasible subsets of B.

Theorem 2.2: Let F be a maximum-weighted feasible subset of B. Then,

$$w(I) \geq \frac{\min_{a\in F} d(a)}{\beta+2} w(F).$$

Proof. Consider a shortest schedule $\mathcal{S}$ of F. Let $\mathcal{C}$ be the collection of compatible sets of requests in F transmitting concurrently in $\mathcal{S}$, and for each $C \in \mathcal{C}$ let $x(C)$ be the total transmission time by C in $\mathcal{S}$. Then,

$$\textstyle\sum_{C\in\mathcal{C}} x(C) \leq 1,$$

and for each $b \in F$,

$$d(b) = \textstyle\sum_{C\in\mathcal{C}} x(C) |C \cap \{b\}|.$$

Thus,

$$\begin{aligned}
w(F) &= \sum_{a\in F} w(a) = \sum_{a\in F} \frac{w(a)}{d(a)} d(a) \\
&= \sum_{a\in F} \frac{w(a)}{d(a)} \textstyle\sum_{C\in\mathcal{C}} x(C) |C \cap \{a\}| \\
&= \textstyle\sum_{C\in\mathcal{C}} x(C) \displaystyle\sum_{a\in F} \frac{w(a)}{d(a)} |C \cap \{a\}| \\
&= \textstyle\sum_{C\in\mathcal{C}} x(C) \displaystyle\sum_{a\in C} \frac{w(a)}{d(a)} \\
&\leq \frac{1}{\min_{a\in F} d(a)} \textstyle\sum_{C\in\mathcal{C}} x(C) \displaystyle\sum_{a\in C} w(a) \\
&\leq \frac{1}{\min_{a\in F} d(a)} \textstyle\sum_{C\in\mathcal{C}} x(C) w(C) \\
&\leq \frac{\max_{C\in\mathcal{C}} w(C)}{\min_{a\in F} d(a)} \textstyle\sum_{C\in\mathcal{C}} x(C) \\
&\leq \frac{\max_{C\in\mathcal{C}} w(C)}{\min_{a\in F} d(a)}.
\end{aligned}$$

Hence,

$$\max_{C\in\mathcal{C}} w(C) \geq w(F) \min_{a\in F} d(a).$$

By Theorem 2.1

$$w(I) \geq \frac{\max_{C\in\mathcal{C}} w(C)}{\beta+2} \geq \frac{\min_{a\in F} d(a)}{\beta+2} w(F).$$

So, the theorem holds. ■

Finally, we remark that the smallest-known β is $\mathcal{O}(\log\alpha)$ [14] when the power assignment is monotone and sub-linear. Thus, in the remaining of this paper, β is assumed to be $\mathcal{O}(\log\alpha)$. Correspondingly, $\mathcal{A}^*$ is an $\mathcal{O}(\log\alpha)$-approximate algorithm for the problem **MWCS**. If all requests in B have demands at least $\Omega(1/\log\alpha)$, then $\mathcal{A}^*$ is also an $\mathcal{O}(\log^2\alpha)$-approximate algorithm for the problem **MWFS**.

III. Transmission Scheduling

In this section, we present a transmission scheduling algorithm which produces a schedule with an explicit bound on the schedule length. Such explicit upper bound is very essential to the joint optimization of selection and scheduling problem. This algorithm has an approximation bound $O(\log^2\alpha)$. We remark that an $O(\log\alpha)$-approximation bound can be achieved by a general approximation-preserving reduction [14] to the problem **MWCS** via a very impractical ellipsoid method. However, the transmission schedule produced by such reduction does not admit any explicit upper bound on the schedule length, and hence cannot be utilized by the joint selection and scheduling algorithm to be presented later in this paper. In contrast, our algorithm is greedy in nature, and is much more efficient in implementation.

A. Conflict Factors

The independence under the SINR model has the following convenient characterization. For two requests a and b in A, the *relative interference* of a toward b, denoted by $RI(a,b)$, is defined as follows: If a and b share a common endpoint (i.e., a and b have a *primary* conflict), then $RI_p(a,b) = \infty$; otherwise,

$$RI(a,b) = \sigma \frac{p(a)\,\ell(a,b)^{-\kappa}}{(p(b) - p_0(b))\,\ell(b)^{-\kappa}}.$$

The *conflict factor* of a request a toward another request $b \in A$, denoted by $\rho(a,b)$, is defined as follows:

$$\rho(a,b) = \min\{1, RI(a,b)\}.$$

Then, a set I of requests is independent if and only if

$$\max_{a\in I} \sum_{b\in I\setminus\{a\}} \rho(I \setminus \{a\}, a) < 1.$$

B. Inductively Independent Set

For any pair of requests a and b, define

$$\overline{\rho}(a,b) = \rho(a,b) + \rho(b,a),$$
$$\widehat{\rho}(a,b) = \min\{1, 2\overline{\rho}(a,b)\}.$$

A subset J of A is said to be *inductively independent* in $\prec$ if

$$\max_{a\in J} \widehat{\rho}(J_{\prec a}, a) < 1.$$

Equivalently, a subset J of A is inductively independent in $\prec$ if and only if

$$\max_{a\in J} \overline{\rho}(J_{\prec a}, a) < 1/2.$$

In general, inductive independence does not imply independence. However, the following facts on an inductively independent subset J of A can be easily verified:

- If $|J| \leq 3$, then J is independent.
- Let
$$I = \{a \in J : \rho(J \setminus \{a\}, a) < 1\}.$$
Then, I is independent and $|I| > |J|/2$.

The above two facts have two implications. First, each inductively independent subset has size less than 2α. Second, a partition Π of an inductively independent subset J into independent sets can be produced as follows.

- Π is initially empty
- While J is not independent, let
$$I = \{a \in J : \rho(J \setminus \{a\}, a) < 1\},$$
add I to Π, and remove I from J.
- Finally, if J is non-empty, add J to Π.

Such partition is referred to the *greedy IS-partition* of J. We claim that
$$|\Pi| \leq 1 + \lfloor \log \alpha \rfloor.$$
Indeed, if J is independent then $|\Pi| = 1$ and the inequality holds. So we assume that J is not independent. Then, $|J| \geq 4$. It is easy to show that
$$|\Pi| \leq \lfloor \log |J| \rfloor \leq \lfloor \log (2\alpha) \rfloor = 1 + \lfloor \log \alpha \rfloor.$$

C. Inductively Compatible Set

A set C of requests is said to be *inductively compatible* in $\prec$ if no pair of requests in C have a primary conflict and
$$\max_{a \in C} \widehat{\rho}(C_{\prec a}, a) < \lambda.$$
Suppose that C is an inductively compatible subset of A in $\prec$. Then C can be greedily partitioned into λ inductively independent subsets
$$\{J_k : k \in [\lambda]\}.$$
in $\prec$ as follows:

- Initially, J_k is empty for each $k \in [\lambda]$.
- For each $a \in C$ in $\prec$, there must exists $k \in [\lambda]$ such that $\widehat{\rho}(J_k, a) < 1$ since
$$\sum_{k \in [\lambda]} \widehat{\rho}(J_k, a) = \widehat{\rho}(C_{\prec a}, a) < \lambda.$$
Pick the first such k and add a to J_k.

Clearly, each J_k for $k \in [\lambda]$ remains inductively independent in $\prec$ throughout the process. The final partition
$$\{J_k : k \in [\lambda]\}.$$
is called a *greedy IIS-partition* of J_k.

In general, inductive compatibility does not imply compatibility. An inductively compatible set C in $\prec$ can be partitioned into at most $1 + \lfloor \log \alpha \rfloor$ compatible subsets as follows.

- Compute a greedy IIS-partition of C into inductively independent sets
$$\{J_k : k \in [\lambda]\}.$$
- For each $k \in [\lambda]$, compute a greedy IS-partition of J_k into independent sets
$$\{I_{k,j} : j \in [l_k]\}$$
- Let $l^* = \max_{k \in [\lambda]} l_k$, and $I_{kj} = \emptyset$ for each $k \in [\lambda]$ and each $l_k < j \leq l^*$. For each $j \in [l^*]$, let C_j^* be the compatible set formed by the λ independent sets I_{kj} for $k \in [\lambda]$.

The final partition
$$\{C_j^* : j \in [l^*]\}$$
is called a *greedy CS-partition* of C.

Inductive compatibility can be conveniently characterized by a *channel-aware* conflict factor function θ defined as follows: For any pair of requests a and b, if a and b are identical or have primary conflict, then
$$\theta(a, b) = 1;$$
otherwise
$$\theta(a, b) = \frac{1}{\lambda} \widehat{\rho}(a, b).$$
Then, a set C is inductively compatible in $\prec$ if and only if
$$\max_{a \in C} \theta(C_{\prec a}, a) < 1.$$

Given a subset S of A, an inductively compatible subset C of S in $\prec$ can be greedily constructed as follows. Initially C is empty. For each $a \in S$ in $\prec$, if $\theta(C, a) < 1$ then add a to C. The final set C is referred to as the *maximal inductively compatible subset* of S in $\prec$. It is maximal in the sense that for each $a \in S$,
$$\theta(C_{\preceq a}, a) \geq 1.$$

D. Inductive Schedule

An *inductive schedule* of a subset $F \subseteq A$ in $\prec$ is a sequence $\mathcal{S}$ of pairs (C_j, x_j) for $1 \leq j \leq k$ where each C_j is an inductively compatible subset in $\prec$ and x_j is a positive number satisfying that for each $a \in F$,
$$d(a) = \sum_{j=1}^{k} x_j |\{a\} \cap C_j|;$$
the value $\sum_{j=1}^{k} x_j$ are referred to as the *length* of $\mathcal{S}$, and is denoted by $\|\mathcal{S}\|$. Any inductive schedule $\mathcal{S}$ of F can be expanded as follows to a schedule of F with the length increased by a factor of at most $1 + \lfloor \log \alpha \rfloor$. For each pair (C, x) in $\mathcal{S}$,

- compute a greedy CS-partition of C in $\prec$ into compatible sets
$$\left\{C_j^* : j \in [l(C)]\right\}$$
for some positive integer $l(C) \leq 1 + \lfloor \log \alpha \rfloor$;
- then replace (C, x) by $l(C)$ pairs
$$\left(C_j^*, x\right) : j \in [l(C)] \,.$$

Let $\mathcal{S}^*$ be the resulting schedule, and it is referred to as the *greedy expansion* of $\mathcal{S}$ in $\prec$. Clearly,
$$\|\mathcal{S}^*\| \leq (1 + \lfloor \log \alpha \rfloor) \|\mathcal{S}\|$$

Given a subset $F \subseteq A$, an inductive schedule $\mathcal{S}$ of F can be produced in the following iterative manner. Initially, $\mathcal{S}$ is empty, let F' be the subset of requests $a \in F$ with $d(a) > 0$. Repeat the following iteration while F' is non-empty:

- Compute a maximal inductively compatible subset C of F' in $\prec$.
- Let $x = \min_{a \in C} d(a)$, and add (C, x) to $\mathcal{S}$.
- For each $a \in C$, replace $d(a)$ by $d(a) - x$, and if $d(a) = 0$ then remove a from F'.

The final $\mathcal{S}$ is referred to as the *greedy inductive schedule of of F in* $\prec$. Since $|F'|$ strictly decreases, the number of iterations is bounded by $|F|$. The next theorem gives an upper bound on the length of $\mathcal{S}$. Denote
$$\Delta(F) := \max_{a \in F} \sum_{b \in F_{\preceq a}} \theta(b, a) \, d(b) \,.$$

Lemma 3.1: $\|\mathcal{S}\| \leq \Delta(F)$.

Proof. Let $\mathcal{C}$ be the collection of inductively compatible sets of requests in F transmitting concurrently in $\mathcal{S}$. For each $C \in \mathcal{C}$, let $x(C)$ be the transmission time by C. Consider an arbitrary request $a \in F$ which completes its transmission last. Then, for each $C \in \mathcal{C}$,
$$\theta(C_{\preceq a}, a) \geq 1.$$
Thus,
$$\begin{aligned}
&\sum_{b \in F_{\preceq a}} \theta(b, a) \, d(b) \\
&= \sum_{b \in F_{\preceq a}} \theta(b, a) \textstyle\sum_{C \in \mathcal{C}} x(C) \, |C \cap \{b\}| \\
&= \textstyle\sum_{C \in \mathcal{C}} x(C) \displaystyle\sum_{b \in F_{\preceq a}} \theta(b, a) \, |C \cap \{b\}| \\
&= \textstyle\sum_{C \in \mathcal{C}} x(C) \displaystyle\sum_{b \in C_{\preceq a}} \theta(b, a) \\
&= \textstyle\sum_{C \in \mathcal{C}} x(C) \, \theta(C_{\preceq a}, a) \\
&\geq \textstyle\sum_{C \in \mathcal{C}} x(C) \\
&= \|\mathcal{S}\|
\end{aligned}$$
So, the lemma holds. ■

E. Greedy Schedule

Now, are ready to present the transmission scheduling of a given subset $F \subseteq A$. The schedule of F is produced in two steps

1) Compute a greedy inductive schedule $\mathcal{S}$ of F in $\prec$.
2) Compute a greedy expansion $\mathcal{S}^*$ of $\mathcal{S}$.

The schedule $\mathcal{S}^*$ is referred to as the *greedy schedule* of F in $\prec$. The following upper bound on $\|\mathcal{S}^*\|$ is an immediate consequence of Lemma 3.1.

Theorem 3.2: $\|\mathcal{S}^*\| \leq (1 + \lfloor \log \alpha \rfloor) \Delta(F)$.

In the remaining of this subsection, we present a lower bound on minimum schedule length $\chi^*(F)$ of F, from which can immediately obtain an approximation bound of $\mathcal{S}^*$. The *backward local independence number* (BLIN) of A in $\prec$, denoted by μ, is defined to the be maximum value of $\widehat{\rho}(I_{\preceq a}, a)$ over all independent subsets I of A and all requests a in A. It was shown in [11] that $\mu = \mathcal{O}(\log \alpha)$. Denote
$$\mu_\lambda := \mu + 2\left(1 - \frac{1}{\lambda}\right).$$

Theorem 3.3: For any $F \subseteq A$ and any $a \in A$,
$$\chi^*(F) \geq \frac{\Delta(F)}{\mu_\lambda}.$$

Proof. We first claim that for any compatible subset $C \subseteq A$ and any $a \in A$.
$$\theta(C_{\preceq a}, a) \leq \mu_\lambda.$$
Let C' be the set of requests in $C_{\preceq a}$ which do not share any common endpoint from a, and C'' be the set of the rest requests in $C_{\preceq a}$. Then,
$$\theta(C_{\preceq a}, a) = |C''| + \frac{1}{\lambda}\widehat{\rho}(C', a) \,.$$
Clearly,
$$|C''| \leq 2.$$
$$|C''| + \widehat{\rho}(C', a) \leq \lambda\mu.$$
Thus,
$$\begin{aligned}
\theta(C_{\preceq a}, a) &= |C''| + \frac{1}{\lambda}\widehat{\rho}(C', a) \\
&= \left(1 - \frac{1}{\lambda}\right)|C''| + \frac{1}{\lambda}\left(|C''| + \widehat{\rho}(C', a)\right) \\
&\leq \left(1 - \frac{1}{\lambda}\right)2 + \frac{1}{\lambda}\lambda\mu \\
&= \mu + 2\left(1 - \frac{1}{\lambda}\right) \\
&= \mu_\lambda.
\end{aligned}$$
So, the claim holds.

Next, we prove the inequality in the theorem. Consider a shortest schedule of F. Let $\mathcal{C}$ be the collection of compatible sets of requests in F appearing in this shortest schedule, and for each $C \in \mathcal{C}$ let $x(C)$ be the total transmission time by C in this shortest schedule. Then,

$$\textstyle\sum_{C\in\mathcal{C}} x(C) = \chi^*(F).$$

and for each $b \in F$,

$$d(b) = \textstyle\sum_{C\in\mathcal{C}} x(C)\,|C \cap \{b\}|.$$

Thus,

$$\begin{aligned}
&\sum_{b\in F_{\preceq a}} \theta(b,a)\, d(b) \\
&= \sum_{b\in F_{\preceq a}} \theta(b,a) \textstyle\sum_{C\in\mathcal{C}} x(C)\,|C\cap\{b\}| \\
&= \textstyle\sum_{C\in\mathcal{C}} x(C) \displaystyle\sum_{b\in F_{\preceq a}} \theta(b,a)\,|C\cap\{b\}| \\
&= \textstyle\sum_{C\in\mathcal{C}} x(C) \displaystyle\sum_{b\in C_{\preceq a}} \theta(b,a) \\
&= \textstyle\sum_{C\in\mathcal{C}} x(C)\,\theta(C_{\preceq a}, a) \\
&\le \mu_\lambda \textstyle\sum_{C\in\mathcal{C}} x(C) \\
&\le \mu_\lambda \chi^*(F).
\end{aligned}$$

So, the theorem holds. ■

The above two theorems immediately imply that the greedy schedule has an approximation bound

$$\mu_\lambda (1 + \lfloor \log \alpha \rfloor) = \mathcal{O}\left(\log^2 \alpha\right).$$

IV. Constrained Inductively Feasible Set

For any $\delta > 0$, a subset F of requests is said to be a δ-*constrained inductively feasible subset* (δ-CIFS) in $\prec$ if $\Delta(F) \le \delta$. By Theorem 3.2, when δ is sufficiently small, any δ-CIFS is feasible. Suppose that B is a set of requests in which the demand of each request is at most $\delta/2$. In this section, we present an algorithm **LR-CIFS** which selects a δ-CIFS F from B based on the local-ratio strategy, and provide a lower bound on $w(F)$ in terms of the maximum weight of *feasible* subsets of B.

We define yet another *demand-aware* conflict factor function τ on B. For any pair of requests a and b in B, if $a = b$ then $\tau(a,b) = 1$; otherwise

$$\tau(a,b) = \frac{d(a)}{\delta - d(b)} \theta(a,b).$$

It is easy to verify that a subset F of B is a δ-constrained IFS if and only if

$$\max_{a\in F} \tau(F_{\prec a}, a) \le 1.$$

Given a subset $S \subseteq B$, a δ-CIFS $F \subseteq S$ can be computed in a *greedy* manner as follows:

- Initially, F is empty.
- For each $a \in S$ in $\prec$, a is added to F if and only if $\tau(F,a) \le 1$.

The final set F is referred to as the *maximal* δ-*CIFS* of S in $\prec$. It is maximal in the sense that for each $a \in S \setminus F$,

$$\tau(F_{\prec a}, a) > 1.$$

Based on local-ratio strategy, a "candidate" subset S is selected from B in a greedy manner as follows.

- Initially, S is empty.
- For each $a \in B$ in the *reverse* order of $\prec$, a discounted weight $\overline{w}(a)$ of a is computed by
$$\overline{w}(a) = w(a) - \sum_{b\in S} \tau(a,b)\,\overline{w}(b);$$
and if $\overline{w}(a) > 0$, a is added to S.

The final set S is referred to as the the *greedy candidate subset* of B in $\prec$.

Now, we are ready to describe our algorithm **LR-CIFS**. The algorithm proceeds in two steps:

- **Step 1**: Compute the greedy candidate subset S of B in $\prec$.
- **Step 2**: Compute the maximal δ-CIFS F of S in $\prec$ and return F.

The theorem below presents a lower bound on the weight of F.

Theorem 4.1: Let O be a maximum-weighted feasible subset of B. Then, $w(F) \ge \frac{\delta}{2\mu_\lambda} w(O)$.

Proof. Let S be the greedy candidate set of B computed at **Step 1**. We shall show that

$$w(F) \ge \overline{w}(S) \ge \frac{\delta}{2\mu_\lambda} w(O),$$

from which the theorem holds.

We first claim that for any subset B of B

$$w(B) = \overline{w}(B) + \sum_{a\in S} \overline{w}(a)\,\tau_k(B_{\prec a}, a).$$

Using the relations between the original weights and the discounted weights, we have

$$\begin{aligned}
w(B) &= \sum_{b\in B} w(b) \\
&= \sum_{b\in B} \overline{w}(b) + \sum_{b\in B} \sum_{a\in S_{\succ b}} \tau_k(b,a)\,\overline{w}(a) \\
&= \sum_{b\in B} \overline{w}(b) + \sum_{b\in B} \sum_{a\in S_{\succ b}} \tau_k(b,a)\,\overline{w}(a) \\
&= \overline{w}(B) + \sum_{a\in S} \overline{w}(a) \sum_{b\in B_{\prec a}} \tau_k(b,a) \\
&= \overline{w}(B) + \sum_{a\in S} \overline{w}(a)\,\tau_k(B_{\prec a}, a),
\end{aligned}$$

where the second inequality follows from the fact that for any pair of requests a and b, $b \in B$ and $a \in S_{\succ b}$ if and only if $a \in S$ and $b \in B_{\prec a}$.

Now, we prove

$$w(F) \geq \overline{w}(S).$$

By the greedy selection of F, for each $a \in S \setminus F$,

$$\tau(F_{\prec a}, a) > 1.$$

Thus,

$$\begin{aligned} w(F) &= \overline{w}(F) + \sum_{a \in S} \overline{w}(a)\, \tau(F_{\prec a}, a) \\ &\geq \overline{w}(F) + \sum_{a \in S \setminus F} \overline{w}(a)\, \tau(F_{\prec a}, a) \\ &\geq \overline{w}(F) + \sum_{a \in S \setminus F} \overline{w}(a) \\ &= \overline{w}(S). \end{aligned}$$

Next, we prove that for any $a \in B$,

$$\tau(O_{\preceq a}, a) \leq \frac{2\mu_\lambda}{\delta}.$$

By Theorem 3.3,

$$\begin{aligned} \mu_\lambda &\geq \sum_{b \in F_{\preceq a}} \theta(b, a)\, d(b) \\ &= d(a)\, |F \cap \{a\}| + \sum_{b \in F_{\prec a}} \theta(b, a)\, d(b) \\ &= d(a)\, |F \cap \{a\}| + (\delta - d(a)) \sum_{b \in F_{\prec a}} \tau(b, a) \\ &= d(a)\, |F \cap \{a\}| + (\delta - d(a))\, \tau(F_{\prec a}, a) \end{aligned}$$

Thus,

$$\begin{aligned} \tau(F_{\prec a}, a) &\leq \frac{\mu_\lambda - d(a)\, |F \cap \{a\}|}{\delta - d(a)} \\ &= \frac{\mu_\lambda - |F \cap \{a\}|\, \delta}{\delta - d(a)} + |F \cap \{a\}| \\ &\leq \frac{\mu_\lambda - \delta\, |F \cap \{a\}|}{\delta - \delta/2} + |F \cap \{a\}| \\ &= \frac{2\mu_\lambda}{\delta} - |F \cap \{a\}| \end{aligned}$$

So,

$$\tau(F_{\preceq a}, a) = |F \cap \{a\}| + \tau(F_{\prec a}, a) \leq \frac{2\mu_\lambda}{\delta}.$$

Finally, we prove

$$\overline{w}(S) \geq \frac{\delta}{2\mu_\lambda} w(O).$$

Since $\overline{w}(a) \leq 0$ for each $a \in B \setminus S$, we have

$$\overline{w}(O) \leq \overline{w}(O \cap S).$$

Thus

$$\begin{aligned} w(O) &= \overline{w}(O) + \sum_{a \in S} \overline{w}(a)\, \tau(O_{\prec a}, a) \\ &\leq \overline{w}(O \cap S) + \sum_{a \in S} \overline{w}(a)\, \tau(O_{\prec a}, a) \\ &= \sum_{a \in S} \overline{w}(a) \left[|O \cap \{a\}| + \tau(O_{\prec a}, a) \right] \\ &= \sum_{a \in S} \overline{w}(a)\, \tau(O_{\preceq a}, a) \\ &\leq \frac{\delta}{2\mu_\lambda} \sum_{a \in S} \overline{w}(a) \\ &= \frac{\delta}{2\mu_\lambda} \overline{w}(S). \end{aligned}$$

So,

$$\overline{w}(S) \geq \frac{\delta}{2\mu_\lambda} w(O).$$

This completes the proof of the theorem. ■

V. Joint Selection & Scheduling

In this section, we present an $\mathcal{O}\left(\log^2 \alpha\right)$-approximation algorithm **DC-JSS** for the problem **MWFS**. We first give a brief overview of the algorithm design strategy. For any positive integer k, we partition A into a "low-demanded" subset

$$A_k := \left\{ a \in A : d(a) \leq \frac{1}{2k} \right\}$$

and a "high-demanded" subset $A'_k := A \setminus A_k$. The algorithm **DC-JSS** will first find a smallest k such that for the $1/k$-CIFS F of A_k computed by the **LR-CIFS** developed in Section IV, its greedy schedule length is at most one and hence it is feasible. Such k must be no more than $1 + \lfloor \log \alpha \rfloor$ by Theorem 3.2. The algorithm **DC-JSS** will then find a compatible (thus feasible) subset C of A'_k can be computed by the algorithm $\mathcal{A}^*$ developed in Section II. The better (in terms of weight) one between F and C is returned as the output. Thus, while the algorithm **DC-JSS** follows the divide-and-conquer strategy, the division component is subtle and it is integrated with one of conquer components. Such subtleness is necessitated due to the NP-completeness of determining the value of α.

Now, we describe our algorithm **DC-JSS**. The algorithm proceeds in three stages:

- **Stage 1**: Initially k is one. Repeat the following iteration which takes three steps:
  - **Step 1**: Compute the maximal $1/k$-CIFS F of A_k by applying the algorithm **LR-CIFS**.
  - **Step 2**: Compute a greedy schedule $\mathcal{S}^*$ of F in $\prec$.
  - **Step 3**: If $\|\mathcal{S}^*\| > 1$, increase k by one, and move on to **Step** 1; otherwise, move on to **Stage 2**.
- **Stage 2**: Compute a compatible subset C of A'_k by applying the algorithm $\mathcal{A}^*$.

- **Stage 3**: If $w(F) > w(C)$ then return F and $\mathcal{S}^*$; otherwise return C.

By Theorem 3.2, **Stage 1** takes at most $1 + \lfloor \log \alpha \rfloor$ iterations.

Next, we establish the approximation bound of the algorithm **DC-JSS**.

Theorem 5.1: The algorithm **DC-JSS** produces an $\mathcal{O}\left(\log^2 \alpha\right)$-approximate solution.

Proof. Let k be the number of iterations taken by **Stage 1**. Then,

$$k \leq 1 + \lfloor \log \alpha \rfloor .$$

Let O_k and O'_k be the maximum-weighted feasible subset of A_k and A'_k respectively. By Theorem 4.1,

$$w(O_k) \leq 2k\mu_\lambda w(F).$$

By Theorem 2.2,

$$w(O'_k) \leq 2k(\beta + 2) w(C).$$

Let O be a maximum-weighted subset of A. Then,

$$\begin{aligned} w(O) &= w(O \cap A_k) + w(O \cap A'_k) \\ &\leq w(O_k) + w(O'_k) \\ &\leq 2k\mu_\lambda w(F) + 2k(\beta + 2) w(C) \\ &\leq 2k(\mu_\lambda + \beta + 2) \max\{w(F), w(C)\} \\ &= \mathcal{O}\left(\log^2 \alpha\right) \max\{w(F), w(C)\}. \end{aligned}$$

So, the theorem holds. ■

VI. Conclusion

The problem **MWFS** involves both selection and transmission scheduling of a feasible subset of requests. A rich set of algorithm design strategies are exploited in our approximation algorithms for this problem. The transmission scheduling follows the *greedy* strategy. At the top level of the selection algorithm, a special *divide-and-conquer* scheme is adopted. At the medium level of the selection algorithm, a *restriction* strategy is applied: the selection of feasible low-demanded requests is restricted to constrained inductively feasible sets, while the selection of feasible high-demanded requests is restricted to compatible subsets. At the bottom level of the selection algorithm, innovative variants of the *local-ratio* (or equivalently, *primal-dual*) scheme are utilized. All our algorithm design and analyses are quite general. Indeed, the dependence on the monotone and sub-linear power assignment is limited to the $\mathcal{O}(\log \alpha)$ upper bound on the two parameters μ and β. We expect that the algorithmic results can be applied to other joint selection and scheduling problems by choosing proper conflict factors among the requests.

We conclude this paper with two open questions. The first open question is with whether the problem **MWFS** can be approximated within $\mathcal{O}(\log \alpha)$ factor with *uniform* power assignment. The second open question is whether the problem **MWFS** can be approximated within a *constant* factor with *linear* power assignment [15] (i.e., the wanted signal strength of each request in a constant). For both open problems, positive answers must take an algorithmic approach dramatically different from that taken in this paper.

ACKNOWLEDGEMENTS: This work was supported in part by the National Natural Science Foundation of P. R. China under grants 61529202, 61572277, and 61722210, by the National Science Foundation of USA under grants CNS-1526638, CNS-1454770, and CNS-1247944.

References

[1] K. Akcoglu, J. Aspnes, B. DasGupta, and M-Y. Kao, Opportunity Cost Algorithms for Combinatorial Auctions, *Applied Optimization 74: Computational Methods in Decision-Making, Economics and Finance*, 455-479, 2002.

[2] M. Andrews and M. Dinitz, Maximizing Capacity in Arbitrary Wireless Networks in the SINR Model: Complexity and Game Theory, *Proc. IEEE INFOCOM* 2009: 1332-1340.

[3] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor and B. Schieber, A Unified Approach to Approximating Resource Allocation And Scheduling, *Journal of ACM* 48(5): 1069-1090, 2001.

[4] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber, Approximating The Throughput of Multiple Machines In Real-Time Scheduling, *SIAM Journal on Computing* 31(2): 331-352, 2001.

[5] R. Bar-Yehuda and D. Rawttz, On The Equivalence between The Primal-Dual Schema And The Local Ratio Technique, *SIAM J. Disc. Math.* 19: 762–797, 2005.

[6] R. Bar-Yehuda and D. Rawttz, Using Fractional Primal-Dual to Schedule Split Intervals with Demands, *Journal Discrete Optimization* 3(4): 275-287, 2006.

[7] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[8] O. Goussevskaia, Y.A. Oswald, and R. Wattenhofer, Complexity in geometric SINR, *Proc. ACM MOBIHOC*2007: 100–109.

[9] M. M. Halldórsson and P. Mitra, Wireless capacity with oblivious power in general metrics,*Proc. SIAM SODA* 2011: 1538-1548.

[10] T. Kesselheim, A Constant-Factor Approximation for Wireless Capacity Maximization with Power Control in the SINR Model,*Proc. SIAM SODA* 2011: 1549-1559.

[11] T. Kesselheim, Approximation Algorithms for Spectrum Allocation and Power Control in Wireless Networks, PhD Thesis, RWTH Aachen University, 2012.

[12] C. Ma, F. Al-dhelaan, and P.-J. Wan. Maximum Independent Set of Links with a Monotone and Sublinear Power Assignment. *Proc. WASA* 2013: 64-75.

[13] P.-J. Wan, Joint Selection And Transmission Scheduling of Point-to-Point Communication Requests in Multi-Channel Wireless Networks, *Proc. ACM MOBIHOC* 2016: 231-240.

[14] P.-J. Wan, O. Frieder, X. Jia, F. Yao, X.-H. Xu, S.-J. Tang, Wireless Link Scheduling under Physical Interference Model, *Proc. IEEE INFOCOM* 2011: 838-845.

[15] P.-J. Wan, L. Wang, C. Ma, Z. Wang, B. Xu, and M. Li, Maximizing Wireless Network Capacity with Linear Power: Breaking The Logarithmic Barrier, *Proc. IEEE INFOCOM* 2013: 135-139.

[16] P.-J. Wan, H.Q. Yuan, X.H. Jia, J.L. Wang, and Z. Wang, Maximum-Weighted Subset of Communication Requests Schedulable without Spectral Splitting, *Proc. IEEE INFOCOM* 2017.

[17] Y. Ye and A. Borodin, Elimination graphs, *ACM Transactions on Algorithms* 8(2):14:1–14:23, 2012.