This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

The k–Constrained Bipartite Matching Problem: Approximation Algorithms and Applications to Wireless Networks

André Berger
Department of Quantitative Economics
Maastricht University, The Netherlands
Email: a.berger@maastrichtuniversity.nl

James Gross
UMIC Research Centre
RWTH Aachen University, Germany
Email: gross@umic.rwth-aachen.de

Tobias Harks
Institute of Mathematics
Technical University Berlin, Germany
Email: harks@math.tu-berlin.de

***Abstract*—In communication networks, resource assignment problems appear in several different settings. These problems are often modeled by a maximum weight matching problem in bipartite graphs and efficient matching algorithms are well known. In several applications, the corresponding matching problem has to be solved many times in a row as the underlying system operates in a time-slotted fashion and the edge weights change over time. However, changing the assignments can come with a certain cost for reconfiguration that depends on the number of changed edges between subsequent assignments. In order to control the cost of reconfiguration, we propose the k-constrained bipartite matching problem for bipartite graphs, which seeks an optimal matching that realizes at most k changes from a previous matching. We provide fast approximation algorithms with provable guarantees for this problem. Furthermore, to cope with the sequential nature of assignment problems, we introduce an online variant of the k-constrained matching problem and derive online algorithms that are based on our approximation algorithms for the k-constrained bipartite matching problem. Finally, we establish the applicability of our model and our algorithms in the context of OFDMA wireless networks finding a significant performance improvement for the proposed algorithms.**

I. Introduction

The bipartite weighted matching problem is fundamental to resource assignment and has therefore been addressed in many different contexts in telecommunication networks [1]–[4]. Such resource assignment problems are characterized by a set of resources and a set of demands. Each resource might have a different utility for each demand and the goal is typically to find an assignment between the two sets such that the utility is maximized. If a resource can be assigned to at most one demand, this problem can be modeled by bipartite weighted matching. Bipartite weighted matching is a graph-theoretical problem which operates over two (disjoint) sets U and V of vertices. For each vertex of U there exist edges with specific weights to some (or all) of the vertices of set V. The problem is to select a subset of edges such that each vertex from $U \cup V$ is incident with at most one edge (or with exactly one edge in the case of the perfect matching problem), and secondly, the total weight of the selected edges is as large as possible. Thus, a maximum weight bipartite matching delivers an assignment of one resource to one demand such that the total utility is maximized. Furthermore, bipartite weighted matching can be solved efficiently by the successive shortest path algorithm with a complexity of $O\left(n^3\right)$ [5], where $n = \max\{|U|, |V|\}$.

The fact that bipartite weighted matching can be solved efficiently has motivated its applications in cases where the computation time is critical [3], [4], [6], [7]. In wireless networks the set of resources often models the channels and the corresponding utility depends on the channel states. As these states, i.e., the signal-to-noise ratio, changes over time, the resource assignment problem has to be executed faster than the change of the channel states occur. Therefore, efficient algorithms are of interest.

In addition to the necessity of solving the resource allocation problem fast, we consider in this paper a subtle complication that arises in practice. In different applications, the corresponding matching problem has to be solved many times in a row due to the slotted nature of the system. If the system is switched to a new configuration (due to changed edge-weights), reconfiguration costs can occur that have a negative impact on the system performance. Typically, this reconfiguration cost grows with the difference between the matching in the previous slot and the current one.

Motivated by reconfiguration costs, we consider in this paper the *k-constrained* bipartite matching problem. In the k-constrained bipartite matching problem, we are given an initial matching and the objective is to compute a new matching with maximum weight such that no more than k edges are modified with respect to the initial matching. Furthermore, we consider the following generalization of the k-constrained bipartite matching problem: Given a set of successive transmission phases, the goal is to compute matchings with maximum total weight (sum of the weights over the phases) such that every two consecutive matchings differ only in at most k edges. Since edge weights are usually not known beforehand (for example due to the unknown channel variations) this problem has a natural *online* variant that we term the *online* k-constrained bipartite matching problem.

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

A. Illustrating Example

Before we outline our results for the k-constrained bipartite matching problem and the organization of the paper, we motivate the considered mathematical problems in the context of the down-link of an orthogonal frequency division multiple access (OFDMA) system. In such systems, the given bandwidth B is split up into $|J|$ parallel communication channels. Furthermore, there is one transmitter – the base station – which queues data destined for $|I|$ terminals upon transmission. The cell operates in a slotted fashion, i.e., time is split into frames of length T_{f} and each frame features a down-link phase and an up-link phase. Frame lengths are typically in the range of milliseconds. Let us focus only on the down-link transmission direction in the following. Per down-link phase, there are S symbols in total. Due to a varying signal-to-noise ratio per receiver and channel, the amount of bits w_{ij}^t that can be transmitted to receiver i on channel j during frame t varies. Assume that prior to each frame t the transmitter has perfect knowledge of the upcoming channel qualities and, hence, of all values w_{ij}^t.

The goal of the base station is to convey the queued data as fast as possible – maximizing the data rate – but also to consider fairness between the terminals. In order to do so, per frame a resource assignment is performed at the base station distributing the channels among the receivers depending on the channel qualities. In order to balance throughput with fairness, the corresponding assignment problem can be modeled by the rate-adaptive optimization problem [8], which is NP-hard. Thus, several works [3], [6] propose a different model that is based on bipartite weighted matching. Specifically, there is a (previously determined) *number* of channels a_i that each terminal will be assigned for each frame (or at least for the upcoming frame and where all these *allocations* add up to $|J|$). This leads to a bipartite weighted matching problem on the balanced complete bipartite graph $G_n = K_{n,n} = (U \cup V, E_n)$, where the receivers are represented by the set U of the bipartite graph (copying each receiver i a_i times according to its allocation) and the set of channels is given by V. Each channel can only be used for data transmission to one terminal at most. This assignment decision is modeled by a binary decision variable x_{ij}^t.

Finally, prior to the transmission of the payload data, the receivers have to be informed of their current assignments to decode their data correctly. From the S symbols available per down-link phase, the first $\bar{S}$ symbols are reserved for transmission of the signaling information (using a robust modulation/coding scheme to ensure correct reception with a high probability). This number of reserved symbols $\bar{S}$ imposes a restriction on the total amount k of matching changes that can be performed from down-link phase to down-link phase (at a fixed "price" of spending $\bar{S}$ symbols on overhead). This leads directly to the k-constrained bipartite matching problem, and, if we consider multiple consecutive down-link phases, to the online version of it.

B. Contributions and Organization of the Paper

In Section II, we derive a fast approximation algorithm (Algorithm 1) for the k-constrained bipartite matching problem. We prove that Algorithm 1 is a $\frac{\lfloor \frac{k}{2} \rfloor}{k}$-approximation that only performs one maximum weight perfect matching computation on a bipartite graph with $4n - 2k$ vertices, and can thus be implemented with running time $O(n^3)$. Moreover, we also consider a natural algorithm that is based on a Lagrangian approach, which has been proposed by Berger et al. [9] for the more general case of budgeted matching problems. This algorithm computes an almost feasible perfect matching of weight at least $OPT - w_{max}$, where OPT denotes the weight of an optimal solution to the k-constrained bipartite matching problem and w_{max} is the maximum weight of an edge in the instance. The running time, however, is $O(n^6)$.

Next, in Section III, we formally introduce the online k-constrained bipartite matching problem, where the goal is to determine matchings sequentially and in an online fashion, that is, edge weights of future phases are not revealed to the algorithm. We first show that no deterministic online algorithm for the online k-constrained bipartite matching problem can have a competitive ratio better than k/n. We also present an online algorithm that almost achieves this bound and has a competitive ratio of at least $(\lfloor k/2 \rfloor)/n$. Finally, we introduce an approximation algorithm for the k-constrained offline matching problem, where the algorithm has information about all edge weights in advance.

Then, in Section IV, we numerically evaluate the presented algorithms in the context of the down-link of an OFDMA cell. For this type of application we find that the difference between the solution to the unrestricted bipartite matching and algorithms addressing the k-constrained version is no more than 30% in terms of the average sum weight of the matchings, showing that our approximation and online algorithms perform very well in this application. If the net system throughput is considered, these algorithms outperform unrestricted bipartite weighted matching by up to 50% as the overhead from signaling information has to be taken into account.

In Section V, we summarize related work from both, the mathematical and the networking point of view, before we conclude the paper in Section VI. Finally, we note that although we put the mathematical contributions in the context of an OFDMA wireless system, there are more potential application areas of our results, even beyond the area of telecommunications. The k-constrained bipartite matching problem is a special case of the budgeted perfect matching problem, for which, to the best of our knowledge, no approximation algorithms are known. Moreover, its online variant has not been addressed so far (neither in the mathematical literature nor in the networking one, see Section V).

II. Approximation Algorithms for the k-constrained Matching Problem

We start this section with formally defining the *k-constrained bipartite matching problem*. For some integer

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

$n \geq 1$ we consider the balanced complete bipartite graph $G_n = K_{n,n}$ with n vertices in each partite set. The vertex set of G_n consists of two disjoint sets $U = \{u_1, \ldots, u_n\}$ and $V = \{v_1, \ldots, v_n\}$ each of cardinality n, and its edge set $E_n = \{u_i v_j : 1 \leq i, j \leq n\}$ consists of all edges between U and V. For a given perfect matching M^0 in G_n, a weight function $w : E_n \to R_0^+$ and an integer parameter $k \geq 0$, the k-constrained bipartite matching problem is to find a maximum weight perfect matching M in G_n that changes at most k edges of M^0, or more precisely M has to satisfy $|M \cap M^0| \geq n-k$. For ease of presentation we assume that the initial matching $M^0 = \{u_i v_i : 1 \leq i \leq n\}$.

In this section we present two approximation algorithms for the k-constrained bipartite matching problem. For a maximization problem a polynomial time algorithm is called an α-approximation for some $\alpha \in [0, 1]$ if the algorithm computes a solution of weight at least α times the weight of an optimal solution. We first present a conceptually very simple $\frac{\lfloor \frac{k}{2} \rfloor}{k}$-approximation (Algorithm 1) that only needs to execute one maximum weight perfect matching on a modified instance. One of the main ideas is to change the weights of the edges slightly (Step 1) to account for an edge being either in the old matching or not. The weight of an edge which is not in the matching M^0 is reduced by the average weight of its two incident edges from M^0.

Algorithm 1 A $\frac{\lfloor \frac{k}{2} \rfloor}{k}$-approximation for the k-constrained bipartite matching problem.

Require: A complete bipartite graph $G_n = K_{n,n} = (U \cup V, E_n)$ with edge weights $w_{ij} \in R_0^+$, the initial perfect matching $M^0 = \{u_i v_i : 1 \leq i \leq n\}$ and a parameter $k \geq 0$.

Ensure: A perfect matching M of G_n with $|M \cap M^0| \geq n-k$.

1: set $w(M^0) = \sum_{i=1}^{n} w_{ii}$.
2: **for all** $1 \leq i, j \leq n$ **do**
3: set $w'_{ij} = w_{ij} + \frac{w(M^0)}{k} - \frac{w_{ii}+w_{jj}}{2}$
4: **end for**
5: find a maximum weight matching M' w.r.t. w' with at most $l := \lfloor \frac{k}{2} \rfloor$ edges
6: **for all** $1 \leq i \leq n$ such that both u_i and v_i are not matched by M' **do**
7: add $u_i v_i$ to M'
8: **end for**
9: extend M' to a perfect matching of G_n in an arbitrary way
10: **return** M'

Theorem 1. *Algorithm 1 is a $\frac{\lfloor \frac{k}{2} \rfloor}{k}$-approximation for the k-constrained bipartite matching problem.*

Proof: We first show that the algorithm indeed outputs a feasible solution to the k-constrained bipartite matching problem. Let $l := \lfloor \frac{k}{2} \rfloor$ and let M' be the matching computed in Step 5 of the algorithm. Let $I = \{i : v_i \text{ or } w_i \text{ is matched by } M'\}$. By the choice of l we have that $|I| \leq k$. Thus, at least $n-k$ edges of M^0 remain, which are all not adjacent with any edge in M' and are added to M' in Step 7. Thus, M' is a feasible solution.

We now prove the approximation ratio of the algorithm. Let M be any feasible perfect matching in G_n and assume w.l.o.g. that $\{u_i v_i : k+1 \leq i \leq n\} \subseteq M$. Let M^* be the remaining edges, i.e., those edges of M with endpoints in $\{u_i, v_i : 1 \leq i \leq k\}$. We first claim that the original weight of M is equal to the modified weight of edges in M^*. That is, $w'(M^*) = \sum_{u_i v_j \in M^*} \left(w_{ij} + \frac{w(M^0)}{k} - \frac{w_{ii}+w_{jj}}{2} \right) = w(M^*) + k \cdot \frac{w(M_0)}{k} - \sum_{i=1}^{k} w_{ii} = w(M^*) + \sum_{i=k+1}^{n} w_{ii} = w(M)$.

This shows that the given problem is equivalent to that of finding a subset of indices $I \subset \{1, \ldots, n\}$ of size $|I| = k$, such that the weight of a perfect matching on the subgraph of G_n induced by the vertices in $\{u_i, v_i : i \in I\}$ w.r.t. w' is maximized. Now consider an optimal solution OPT to this modified problem. Let L be the set of the l heaviest edges (w.r.t. w') in OPT. The total weight (w.r.t. w') of the edges in L is at least $\frac{\lfloor \frac{k}{2} \rfloor}{k}$ times the weight of OPT. However, the set L is also a feasible solution for the problem solved in Step 5 of the algorithm. Thus, our algorithm finds a solution with weight (again w.r.t. w') at least $\frac{\lfloor \frac{k}{2} \rfloor}{k}$ times the weight of an optimal solution to the modified problem. Hence, it provides a solution for the k-constrained bipartite matching problem of weight at least the same fraction of the optimal solution. ■

Note that the problem in Step 5 can be solved by adding two independent sets of size $n-l$ and connecting all the vertices of the first set to U and all vertices of the second set to V with edges of weight zero and then finding a maximum weight perfect matching in the augmented graph. This also implies that the algorithm can be implemented to run in $O(n^3)$ time.

For even k Algorithm 1 is a 1/2-approximation. If k is odd and small (say $k \leq 1/\epsilon + 1$), then the optimal solution can be found by exhaustive search. On the other hand, if $k > 1/\epsilon + 1$, then $\frac{\lfloor \frac{k}{2} \rfloor}{k} \geq 1/2 - \epsilon$. This implies that Algorithm 1 can be used to devise a $1/2 - \epsilon$-approximation for all k that runs in time $O(n^{(1/\epsilon)})$. Also note that our algorithm works for complete graphs as well without modification, except that for Steps 5 and 9, a maximum weight perfect matching algorithm for general graphs has to be used.

We now turn to a bicriteria algorithm for the k-constrained bipartite matching problem. In fact, this algorithm will work for the more general case of $\{0, 1\}$ cost functions, i.e. with each edge $u_i v_j$ a cost $c_i \in \{0, 1\}$ is associated, and a perfect matching M of G_n is feasible if $c(M) \leq k$ for a given input parameter k. We denote by OPT_k the maximum weight of a perfect matching M with $c(M) \leq k$. As it is often done for bicriteria optimization problems we use as an initial step a Lagrangian relaxation of the integer programming formulation. For simplicity, we will from now on write $PM(G_n) = \{x \in \{0,1\}^{|E|} : \sum_{j=1}^{n} x_{ij} = 1 \text{ for all } 1 \leq i \leq n \text{ and } \sum_{i=1}^{n} x_{ij} = 1 \text{ for all } 1 \leq j \leq n\}$ to denote the set of binary vectors

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

representing the perfect matchings of G_n. The integer program for the constrained bipartite matching problem with $\{0,1\}$ costs is given below.

$$\begin{aligned} \max w \cdot x, \text{ subject to} \\ x \in PM(G_n) \\ c \cdot x \leq k \\ x \in \{0,1\}^{|E|}. \end{aligned}$$

For some real number $\lambda \geq 0$, the Lagrangian relaxation of the above integer program is:

$$\begin{aligned} z(\lambda) = \max(w \cdot x + \lambda(k - c \cdot x)), \text{ subject to:} \\ x \in PM(G_n), \\ x \in \{0,1\}^{|E|}. \end{aligned}$$

For a fixed $\lambda \geq 0$, the problem $LR(\lambda)$ is a maximum weight perfect matching problem with modified Lagrangian weights $w_\lambda(u_i v_j) = w(u_i v_j) - \lambda c(u_i v_j)$ and is solvable in polynomial time. The optimal value $z(\lambda)$ provides an upper bound on the optimal solution and one therefore tries to find $\lambda \geq 0$ such that $z(\lambda)$ is minimized. Finding the optimal λ is called the dual Lagrangian problem, and if the underlying problem without the complicating constraint can be solved in polynomial time, then so can the Lagrangian dual. After having solved the Lagrangian dual, one can obtain two perfect matchings, one of high weight but possibly not satisfying the cost constraint, and one that does satisfy the cost constraint but possibly has low weight. Using a similar approach as for the budgeted matching problem [9, Lemma 3.1], these two solutions can be merged into one with a provable high weight as stated in the following theorem. In fact, for our application in OFDMA wireless networks, we only use the solution obtained from solving the Lagrangian dual as it turns out to be almost optimal.

Theorem 2. *There exists an algorithm which computes a perfect matching M with cost at most $c(M) \leq k+1$ and weight at least $OPT_k - w_{max}$, where w_{max} denotes the maximum weight of an edge in G_n. Using parametric search [10] to solve the Lagrangian dual, the algorithm can be implemented in $O(n^6)$ time.*

Finally, we note that this algorithm can again be implemented with the same approximation guarantees for complete graphs.

III. The Online k-constrained bipartite matching problem

In this section, we introduce an online variant of the k-constrained bipartite matching problem that captures the sequential structure of systems that arise in practice.

An instance of the online k-constrained bipartite matching problem consists of a balanced complete bipartite graph $G_n = K_{n,n} = (V \cup W, E_n)$ with n nodes in each partite set. Moreover, we are given a sequence of edge weights $\sigma = (w^1, \ldots, w^T)$, where $w^t : E_n \to R_0^+, t = 1, \ldots, T$, $T \in \mathbb{N}$. Here, T denotes the number of time slots. The goal is to sequentially calculate perfect matchings M^t, $t = 1, \ldots, T$, such that the total weight $\sum_{t=1}^{T} w^t(M^t)$ is maximized. We make the following three crucial assumptions: (i) edge weights are revealed in an *online fashion*, that is, edge weights are only revealed for the current time slot and future edge weights are not known; (ii) once a matching is determined, no change of this matching is possible; (iii) every matching M^t may have at most k changes with respect to its predecessor matching M^{t-1}, $t = 1, \ldots, T$.

Knowing all edge weights in advance, we call the problem of maximizing the total weight subject to the matching constraints the *offline optimization problem* and denote the offline optimal solution (and its total weight) by OPT.

A. Competitive Analysis

For a given sequence of weights $\sigma = (w^1, \ldots, w^T)$ and a sequence of perfect matchings $(M^1, \ldots, M^T)$ produced by an online algorithm ALG, we denote by $\text{ALG}(\sigma)$ the total weight of all perfect matchings in the output sequence. The online algorithm ALG is called (strictly) *c-competitive*, if for all possible sequences σ, $\text{ALG}(\sigma)$ is never smaller than c times the total weight of an optimal offline solution. The *competitive ratio* of ALG is the supremum over all $c \geq 0$ such that ALG is c-competitive; see for instance Borodin and El-Yaniv [11] and Fiat and Woeginger [12].

We first show that no deterministic online algorithm can achieve a competitive ratio better than k/n. We then complement this bound by presenting an algorithm that actually achieves this bound up to a factor of two, i.e., there exists an online algorithm with competitive ratio $\lfloor k/2 \rfloor / n$.

Theorem 3. *The competitive ratio of any deterministic online algorithm is at most k/n, even when all weights are restricted to be in the set $\{0,1\}$, where n is the size of the balanced complete bipartite input graph G_n.*

Proof: We construct an instance of the k-constrained online matching problem with $T \geq \lfloor n/k \rfloor + 1$ time slots as follows. The initial matching M^0 is any arbitrary perfect matching in G_n.

We specify $\sigma = (w^1, \ldots, w^T)$ as follows. All weights of the first $T-1$ time-slots remain zero, i.e. $w_{ij}^t = 0$ for all $1 \leq t \leq T-1$ and all $1 \leq i, j \leq n$. Let ALG be an arbitrary deterministic online algorithm that determines the matching M^{T-1} in time slot $T-1$. Given M^{T-1}, the online adversary determines the edge weights for time-slot T as

$$w_{ij}^T = \begin{cases} 0 & \text{if } (i,j) \in M^{T-1} \\ 1, & \text{else.} \end{cases}$$

Clearly, ALG achieves for the first $T-1$ time-slots a total weight of 0. In the last phase, ALG can add at most k edges of weight 1, i.e., the total weight for all T phases is at most k. The optimal matching (anticipating the high weight in the last time slot) will be able to add k new edges with weight 1 in every time-slot (possibly less in the last time slot if $k \nmid n$).

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Since there are $T \geq \lfloor n/k \rfloor + 1$ time slots, it can achieve an overall weight of n and thus $\text{ALG}(\sigma) \leq (k/n)\,\text{OPT}$. ■

We proceed with presenting the online algorithm that achieves the upper bound for the competitive ratio up to a factor of two. The idea of the algorithm is similar to the one used in Algorithm 1. Given edge weights w^t in time slot t and a perfect matching M^{t-1}, we first compute a maximum weight matching w.r.t. w^t having at most $\lfloor k/2 \rfloor$ edges. This matching can be extended to a perfect matching having at most k changes from M^{t-1}.

Theorem 4. *The competitive ratio of the above online algorithm is at least $(\lfloor k/2 \rfloor)/n$, where n is the size of the balanced complete bipartite input graph G_n.*

Proof: Let $\sigma = (w^1, \ldots, w^T)$ be arbitrary and consider the two solutions ALG produced by the above algorithm and OPT, the optimal solution for the corresponding offline problems with weight sequence σ. Let OPT^t and ALG^t denote the weight of OPT and ALG in time slot t, respectively. Similarly to the proof of the approximation guarantee of Algorithm 1, we can argue that the $\lfloor k/2 \rfloor$ heaviest edges of OPT^t comprise a feasible solution to the problem that ALG solves in time slot t. Hence, for every $1 \leq t \leq T$ we have that $\text{ALG}^t \geq (\lfloor k/2 \rfloor)/n\,\text{OPT}^t$ and the claimed competitive ratio follows as we sum up this inequality over all time slots. ■

Note that the above algorithm and Algorithm 1 only differ in the weight function that are used and that it is only due to the modification of the original weights that enables us to achieve a $(\lfloor k/2 \rfloor)/k$-approximation. We can also combine the algorithm from Theorem 4 and Algorithm 1 to obtain an online algorithm with competitive ratio of $(\lfloor k/2 \rfloor)/n$, that at the same time provides a $(\lfloor k/2 \rfloor)/k$-approximation to the optimal solution for each time slot.

Corollary 1. *The online algorithm that in each time slot chooses from the solutions of the algorithm from Theorem 4 and of Algorithm 1 the one with higher weight, has competitive ratio at least $(\lfloor k/2 \rfloor)/n$ and provides a $(\lfloor k/2 \rfloor)/k$-approximation for the k-constrained matching problem in each time step.*

B. Approximating the Offline Optimum

In this section, we try to solve the offline problem assuming that an algorithm knows the entire input sequence σ in advance. Note that the offline problem is at least as hard as the k-constrained bipartite matching problem. In fact, formulating the offline problem using the integer formulation yields even a *non-linear* integer program. Note that Corollary 1 implies that the algorithm given therein is also a $(\lfloor k/2 \rfloor)/n$-approximation for the offline problem.

However, if the number of vertices in the bipartite graph is large, this may give us only a small approximation ratio. In turn, for a small number of phases we can provide a better algorithm. We derive an approximation algorithm for the offline problem that is based on approximation algorithms for the k-constrained bipartite matching problem as analyzed in the previous section.

Suppose we have an approximation algorithm $\mathcal{A}$ for the k-constrained bipartite matching problem that achieves an approximation guarantee $\alpha \in [0, 1]$. Before we outline a simple approximation algorithm for the offline problem, we introduce some notation.

We call the tuple $(M^1, \ldots, M^t)$ a feasible matching up to phase t, if every $M^r, r = 1 \ldots, t$ is feasible (at most k changes) with respect to its predecessor. We say that M^{t+1} has *support* if there exists a tuple $(M^1, \ldots, M^t)$ such that $(M^1, \ldots, M^t, M^{t+1})$ is feasible. In this case, we also say $(M^1, \ldots, M^t)$ supports M^{t+1}.

Lemma 1. *Every perfect matching M having at most $t\lfloor \frac{k}{2} \rfloor$ changes ($t \geq 1$) from the initial perfect matching M^0 has support of size $t-1$, that is, there is $(M^1, \ldots, M^{t-1})$ supporting M.*

Proof: Write all changed edges of M in a list $(e_1, \ldots, e_b)$, where $b \leq t\lfloor \frac{k}{2} \rfloor$. At most k edges of M^0 are adjacent with an edge in $\{e_1, \ldots, e_{\lfloor \frac{k}{2} \rfloor}\}$. Therefore, we can construct a perfect matching M^1 containing at least $n-k$ edges from M^0 and all edges in $\{e_1, \ldots, e_{\lfloor \frac{k}{2} \rfloor}\}$. In the same way we can add the edges from $\{e_{\lfloor \frac{k}{2} \rfloor + 1}, \ldots, e_{2\lfloor \frac{k}{2} \rfloor}\}$ to M^1 in order to construct M^2. Note that in this process, at every step, only edges are deleted, which are adjacent to a newly inserted edge. Since the set $\{e_1, \ldots, e_b\}$ comprises a matching itself, no edge e_i that was inserted at an earlier step will ever be deleted later. Thus we can reach the perfect matching M in t steps and by construction the sequence $(M^1, \ldots, M^{t-1})$ supports M. ■

Lemma 2. *Consider the following problems $\max_M w(M)$ subject to (i) M has at most ℓ, $1 \leq \ell \leq p$, changes from some initial matching M^0 and (ii) M has at most p changes from some initial matching M^0. Let $\tilde{M}, \bar{M}$ be the optimal solutions to the above problems, respectively. Then,*

$$w(\tilde{M}) \geq \frac{\lfloor \frac{\ell}{2} \rfloor}{p} w(\bar{M}).$$

Proof: Order the weights of the optimal matching $\bar{M}$ in non-increasing order. Then, the first $\lfloor \frac{\ell}{2} \rfloor$ heaviest weights can be realized in a matching, say M', so that M' has only ℓ changes with respect to M^0. For M', we have $w(M') \geq \frac{\lfloor \frac{\ell}{2} \rfloor}{p} w(\bar{M})$, thus the claim follows. ■

Algorithm 2 A $\Omega\left(\frac{\alpha}{T}\right)$-approximation for the multi-phase k-constrained bipartite matching problem.

1: for every $1 \leq t \leq T$, run $\mathcal{A}$ for problem $\max_{M^t} w^t(M^t)$ subject to at most $t\lfloor \frac{k}{2} \rfloor$ changed edges with respect to M^0 and return the solution $\tilde{M}^t$ and $w^t(\tilde{M}^t)$
2: calculate $\ell = argmax_{t \in [T]}\{w^t(\tilde{M}^t)\}$ and compute a solution $(M^1(\tilde{M}^\ell), \ldots, M^{\ell-1}(\tilde{M}^\ell), \tilde{M}^\ell, \ldots, \tilde{M}^\ell)$, where $M^i(\tilde{M}^\ell), i = 1, \ldots, \ell-1$, is chosen so as to *support* $\tilde{M}^\ell$
3: **return** $(M^1(\tilde{M}^\ell), \ldots, M^{\ell-1}(\tilde{M}^\ell), \tilde{M}^\ell, \ldots, \tilde{M}^\ell)$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

In the next theorem, we establish a bound on the approximation ratio of the above algorithm. For technical reasons, we will assume $k \geq 5$. Note that for $k \leq 4$, one can even prove better bounds.

Theorem 5. *For $k \geq 5$, the algorithm is an $\Omega(\frac{\alpha}{T})$-approximation.*

Proof: We first observe that $\lfloor \frac{k}{2} \rfloor / 2 \geq k/5$ for all $k \geq 5$. Thus for any $t \geq 1$, we have that $\frac{\lfloor t \lfloor \frac{k}{2} \rfloor \rfloor}{2} / (t\,k) \geq \frac{1}{10}$.

Let $(\tilde{M}^1, \ldots, \tilde{M}^T)$ and $(O^1, \ldots, O^T)$ be the matchings of the algorithm and the optimal matchings, respectively. Let $\bar{M}^1, \ldots, \bar{M}^T$ be the matchings that solve for every $t = 1, \ldots, T$, $\bar{M}^t$ the problem

$$\max_M w^t(M) \text{ s.t.: } M \text{ has at most } t\,k \text{ changes from } M^0.$$

Clearly, $w^t(\bar{M}^t) \geq w^t(O^t)$ for all $t = 1, \ldots, T$, since every O^t is a feasible matching to the above problem.

Let $\textsc{Opt} = \sum_{t=1}^{T} w^t(O^t)$. Then, for every t there exists $\lambda_t \in [0,1]$ with $\sum_{t=1}^{T} \lambda_t = 1$ so that we can partition $\textsc{Opt}$ as

$$\lambda_t\, \textsc{Opt} = w^t(O^t),\ t = 1, \ldots, T.$$

Let $\tilde{M}^\ell$ be as in the description of the algorithm. Then, using Lemma 2 with $l = t\lfloor \frac{k}{2} \rfloor$ and $p = tk$, and the above inequality, we get that

$$\begin{aligned} w^\ell(\tilde{M}^\ell) &\geq \max_{t \in [T]} \{ \frac{\alpha}{10}\, w^t(\bar{M}^t) \} \geq \max_{t \in [T]} \{ \frac{\alpha}{10}\, w^t(O^t) \} \\ &\geq \min \max_{\lambda_t \in [0,1],\, t=1,\ldots,T,\, \sum_{t=1}^{T} \lambda_t = 1} \{ \frac{\alpha}{10}\, \lambda_t \textsc{Opt} \}. \end{aligned}$$

The minimum is attained for $\lambda_t = \frac{1}{T}$, $t = 1, \ldots, T$, proving the claim. ∎

The algorithm is well suited for applications in which T is small. Moreover, we note that the algorithm tries to find that phase in which it can achieve the highest weight. In particular, it will be able to solve that instances well that we previously introduced as the worst case for online algorithms (see Theorem 3), in which all weight is concentrated in one phase.

IV. Application and Computational Results

We evaluate the proposed algorithms in the context of down-link data transmission in long term evolution (LTE) OFDMA systems. The underlying system model has already been presented in Section I-A. In the following we first present the system parametrization, then the methodology and finally the results.

A. System Parameters

We consider as system scenario the down-link transmission in 3 GPP LTE OFDMA systems [13], [14]. We assume a bandwidth of $B = 20$ MHz. Subcarrier bandwidth is fixed at 15 kHz. The usual operation assumes a frame of length 10 ms which is divided into 20 subframes. In TDD mode these are alternated between up- and down-link. Due to the low bandwidth of the subcarriers and the length of the guard interval (where we assume it to be set to 4.7 μs), in total this ends up in $S = 7$ symbols used per subframe. Resource allocation in the down-link is based on the definition of a resource block. This is the smallest number of subcarriers/time slots per subframe that can be dynamically assigned. The baseline LTE design specifies that 12 consecutive subcarriers form a resource block, while one such block spans the complete subframe[1]. Hence, the considered system features $|J| = 96$ resource blocks available for transmission. For each subframe the scheduler assigns resource blocks to different terminals based on channel state information provided to the base station. We assume in the following that this channel state information is accurate. The total transmit power is set to 200 W over the entire bandwidth, every resource block receives a fixed transmit power of about 2 W. The noise power is set to -100 dBm. Furthermore, we assume that $|I| = 96$ terminals are in the cell. For each of these terminals the base station has a large amount of data queued waiting for transmission. We consider the system performance over a time span of 20 sec., leading to a total of $T = 2000$ down-link phases.

In order to characterize the channel states of the resource blocks, the following assumptions are made: We consider a cell with a radius of 100 m and terminals are (uniformly distributed over the area). A center frequency of 2 GHz is considered. For path loss we assume a standard model with $10 \log(k) = 69.9$ dB and $\alpha = 3.7$. Log-normal shadowing is assumed and parameterized by $\sigma = 8$ dB. Fading is specified by a Jakes power spectrum with a Doppler shift according to the center frequency and an object velocity of up to 100 m/s. Furthermore we assume an exponential power delay profile with a delay spread of 1 μs. Based on the channel gains generated according to this propagation environment, the resulting SNR per resource block determines the amount of bits that this resource block can carry per resource element. We assume for this relationship Shannon capacity: $\bar{w} = \log_2(1 + \bar{\gamma})$.

In LTE a varying amount of symbols are consumed for the Physical Downlink Control Channels (PDCCH) [14]. For TDD there can be up to 3 symbols used for control information out of 7 symbols per down-link phase in total. The control information includes, among other control elements, a terminal identifier, the assigned resource blocks and the modulation/coding used on these resource blocks. There exist different encodings for these information. We consider the following model for the down-link control channels: To signal the modification of a single resource block assignment it takes 36 resource elements. As there are in total 1152 resource elements per symbol, this allows to signal 32 assignment modifications per symbol. In total we assume that no more than 1 symbol should be used for control information, which sets $k = 32$ in the matching problem. By setting all terminals into semi-persistent scheduling mode with a repetition interval of every down-link phase, every terminal will reuse the assigned resource blocks if there is no new assignment information for

[1]Note that for a frame length of 7 this ends up in a total of 84 OFDM symbols (12 subcarriers times the 7 symbols per subcarrier.). The LTE specification refers to this as 84 resource elements

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

it for the current down-link phase.

B. Metrics and Comparison Schemes

As performance metric we consider the average throughput per terminal in the following. More precisely, we consider the throughput per symbol during payload transmission as primary metric. This results from the weight of a certain (either constrained or unconstrained) matching and simply considers the average weight of one edge in this matching, i.e. $\sum_{\forall t} w\left(M^t\right)/n{\cdot}T$. Furthermore, we consider as second metric the total amount of data that can be transmitted per terminal over the entire considered time span. This metric depends on the net throughput per down-link phase and also takes into account the number of "lost" symbols due to signaling. Formally, it is given by $\sum_{\forall t}\left(S-\lceil\frac{c\left(M^t\right)}{32}\rceil\right)\cdot 12\cdot w\left(M^t\right)/n$. In this formulation, the number 12 stems from the fact that one edge in the matching represents a bundle of 12 channels in LTE (see above).

We consider four different approaches to solve the online k-constrained bipartite matching problem. These different approaches perform per step the following algorithms:

- **Maximum Weight Matching** Here, per frame the solution to the unrestricted weighted matching instance is generated. It serves as comparison scheme. For our considered problem there is no guarantee that it generates a valid assignment as the number of newly assigned resource blocks might be larger than 32. Nevertheless, we can quantify its effective throughput as the larger number of assignments cost more signaling symbols in our model. Note that this approach has the complexity of $O\left(n^3\right)$.
- **Optimal k–Constrained Matching** Per frame the optimal solution to the IP formulation of the k-constrained bipartite matching problem (as formally introduced in Section II) is computed.
- **1/2–Approximation** In this approach, per frame Algorithm 1 is executed. As stated, it represents a viable option to determine the solution to the k-constrained bipartite matching problem and therefore also to its online version. The complexity of this algorithm is $O\left(n^3\right)$ (note that here $k=32$ is even and thus we do have an $1/2$-approximation).
- **Lagrange** Finally, in this approach per frame we solve the Lagrangian dual problem as described in Section II. This represents a feasible solution technique in practice as well but with a much higher complexity of $O\left(n^6\right)$.

Note that (except for the first phase) in this evaluation we do not directly compare the algorithms on the same set of instances, since as we go through the frames the different approaches will compute different matchings for frame t which serves as the input matching for frame $t+1$.

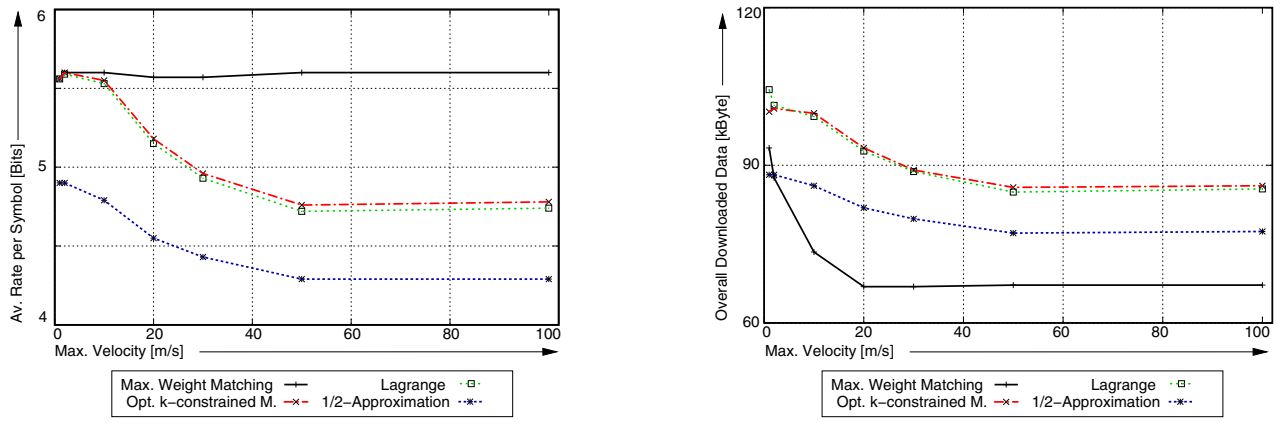
Methodologically, these different approaches are evaluated for different settings of the maximum object velocity in the propagation scenario. The reason for varying the velocity is that with an increasing velocity the correlation of the channel states, i.e. edge weights, between two consecutive instances decreases. The coherence time is a good metric which quantifies this relationship. It indicates for a certain propagation environment the time shift for which two channel states are still correlated by a value of 0.9. We vary the velocity between $1\ \mathrm{m/s}$ up to $100\ \mathrm{m/s}$ which corresponds to a decrease of the coherence time from $33\ \mathrm{ms}$ down to $0.33\ \mathrm{ms}$. As in between each down-link frame there is one up-link frame, this means that for small velocity consecutive edge weights are strongly correlated while for large velocities consecutive edge weights are uncorrelated. For each setting for the maximum velocity of the environment, we generate channel instances from the characterization of the propagation environment mentioned. These channel instances are then processed by C++ programs, in which the appearing linear and mixed integer programs are solved using the CLP and CBC solvers developed within the COIN OR project [15]. The resulting assignments are then used for statistical analysis.

C. Numerical Results

In Figure 1(a) we present the average throughput per symbol (in bits) for the different approaches and the varying velocity setting. First notice that the unrestricted maximum weight matching approach constantly achieves a rate of 5.5 bits per symbol. In comparison, all other approaches, which are subject to the k constraint, have a decreasing performance as the velocity increases. The upper bound to the k-constrained matching initially achieves 5.5 bits per symbol and drops down to 4.8 bits per symbol for high velocities. The Lagrange approach can almost match that performance with a very little performance degradation of about 1% for any velocity setting. The 1/2 approximation features also a constant degradation compared to the optimal k constrained matching approach but the absolute difference is larger with about 0.5 bits per symbol less rate. Notice that the theoretical bound of the 1/2–approximation is much lower, hence, in this application even in the case of uncorrelated consecutive problem instances the 1/2 approximation performs exceptionally well if also taking its significantly lower complexity into account.

In Figure 2 we compare the average number of changed edges per frame for the different approaches. As the velocity increases the unrestricted matching approach converges fast to a very high amount of edge changes. For the highest velocity settings it essentially changes almost every edge from frame to frame. In contrast, all three k-constrained approaches stick to the 32 allowed number of edge changes (as it would be expected) with the direct performance penalty shown in the previous Figure 1(a). Finally, in Figure 1(b) the resulting average amount of downloaded data (in kBytes) is shown for the different approaches. Notice that this takes also the loss due to signaling into account. All variants that address the k-constrained matching problem spend per down-link phase at most one symbol for control information (the Lagrange approach sometimes even changes no edge at all leading to 0 symbols spent for control information). As we know from Figure 2 these three approaches change for medium to high velocities constantly all 32 possible edges and hence, spend

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.



(a) Av. Rate per Symbol

(b) Av. Downloaded Data Amount

Fig. 1. Comparison between the av. rate per symbol and the resulting net amount of downloaded payload data for the four different approaches over a varying maximum velocity in the propagation environment.

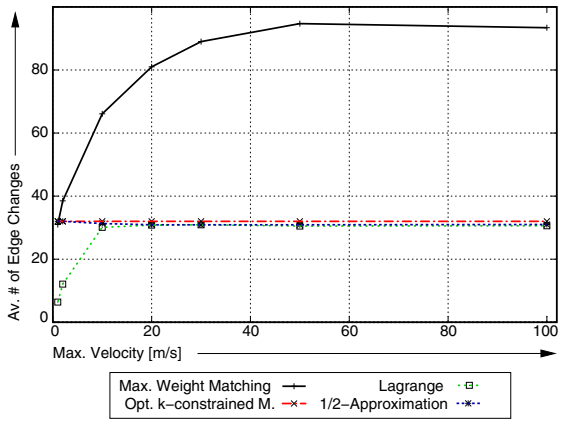


Fig. 2. Average number of changed edges per down-link phase for the four different approaches over a varying maximum velocity in the propagation environment.

one symbol for control information. Thus, the performance of these three approaches in comparison to each other, as observed in Figure 1(b), is equal to the observation in Figure 2. However, for the approach based on the unrestricted matching problem the resulting amount of required control symbols is higher, as many more edges are modified in order to achieve a (slightly) higher rate per symbol. This leads to a performance trade-off, which has been addressed in [16] and is not the focus of this contribution. Nevertheless, the resulting performance would be much lower in the considered system with a performance difference of up to 50% if comparing it to the Lagrange approach. In comparison to the 1/2 approximation, we are still able to provide a performance advantage up to 30% at the same level of computation complexity.

We additionally observe that for all our test instances, the performance of our online algorithm (successively applying the 1/2 approximation) is much better in practice as the provable performance guarantee of $1/6$ (Theorem 4 and Corollary 1) suggest. In fact, using the average amount of downloaded data of the maximum weight matching (assuming that no edge changes are needed), which gives an upper bound for the average amount of downloaded data of the optimal offline solution, we see that our online algorithm has a competitive ratio of more than 80% (on average) for our test instances.

V. Related Work

The k-constrained bipartite matching problem can be considered as a special case of the more general *budgeted perfect matching problem*. In this problem, in addition to a graph G and non-negative weights on its edges, also non-negative costs on the edges as well as a budget B are given. The goal is to find a maximum weight perfect matching in G which has total cost at most B. Given an instance of the k-constrained bipartite matching problem, it can be cast as a budgeted perfect matching problem by letting the cost of edges in M^0 equal 0 and the cost of all other edges equal 1, and by setting $B = k$. The budgeted perfect matching problem is NP-hard and no approximation algorithms for it are known. Recently, a polynomial time approximation scheme was developed for a closely related budgeted matching problem with general weights and costs [9]. Closely related to budgeted perfect matching problem is the *exact perfect matching problem*, as first introduced by Papadimitriou and Yannakakis in 1982 [17]. This problem asks whether in an edge weighted bipartite graph there exists a perfect matching of weight exactly equal to a given weight W. This problem is NP-hard for arbitrary weights, but it has remained one of the intriguing open questions in combinatorial optimization whether it is also NP-hard for polynomially bounded weights. For polynomially bounded weights and costs the budgeted matching problem, the budgeted perfect matching problem and the exact perfect matching problem are all equivalent [9]. Although we do not settle the complexity of the k-constrained bipartite matching problem or its generalization to the budgeted perfect matching problem with costs in $\{0, 1\}$ in this paper, the connection to this long standing open problem may explain this difficulty.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

The online bipartite matching problem also appears in other works starting with results from Karp et al. [18]. However, in that case the vertices of the graph are revealed in an online fashion and thus this problem does not relate to our work. To the best of our knowledge no work has been done on online matching problems where matchings have to be found consecutively coupled by certain constraints.

Bipartite weighted matching problems which model resource allocation problems in telecommunication networks have been addressed, for example, in the context of switching [1], wave-length division multiplexing optical networks [2], OFDMA wireless networks and cognitive (wireless) networks [4], [7]. For OFDMA networks it is proposed mainly to reduce computational complexity of otherwise NP-hard optimization problems [3], [6] and has been found to provide a quite good equivalent performance.

It has been shown that the control information can become a significant drawback in the down-link of OFDMA systems [16], [19]. In order to reduce the signaling overhead, different techniques have been studied. A quadratic optimization model that maximizes net throughput has been proposed in [16]. Here, the number of symbols that the control information can consume per down-link phase is taken as a variable and the dependence between the achieved bit rate during the payload and the required signaling symbols is modeled. Note that a subsequent solution of the k-constrained bipartite matching problem can address this trade-off.

VI. Conclusions

In this paper we address assignment problems with additional constraints that arise from applications in which many subsequent assignments of resources have to be realized over time and in which the cost of reconfiguration between two consecutive assignments can have a significant impact on the overall performance. Furthermore, the run time of applied algorithms is of importance.

To this end we introduce the k-constrained bipartite matching problem and provide fast approximation algorithms with provable approximation guarantees for this problem. We also consider the online version of the problem as it models the problems arising in such time-slotted applications. We provide (almost) tight upper and lower bounds on the competitive ratio of online algorithms for this problem.

Numerically, we can show that in the context of the down-link of OFDMA wireless cells the proposed approximation algorithms provide quite good performance compared to the optimal k-constrained bipartite matching as well as compared to the unrestricted bipartite matching problem. This even holds for weakly correlated problem instances. This good performance leads, as a side-effect, to a reduction of the overhead due to signaling in such systems and improves therefore overall system throughput.

Acknowledgment

This research was funded in part by the DFG Cluster of Excellence on Ultra-high Speed Information and Communication (UMIC), German Research Foundation grant DFG EXC 89.

References

[1] M. Goudreau, S. Kolliopoulos, and S. Rao, "Scheduling algorithms for input-queued switches: randomized techniques and experimental evaluation," in *Proceedings IEEE INFOCOM*, vol. 3, March 2000, pp. 1634 – 1643.

[2] Z. Zhang and Y. Yang, "Optimal scheduling algorithms in wdm optical interconnects with limited range wavelength conversion capability," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 1012–1026, Nov 2004.

[3] H. Yin and H. Liu, "An efficient multiuser loading algorithm for OFDM-based broadband wireless systems," in *Proc. of IEEE Globecom*, November 2000.

[4] C. Zhao, M. Zou, B. Shen, B. Kim, and K. Kwak, "Cooperative spectrum allocation in centralized cognitive networks using bipartite matching," in *Proceedings IEEE Global Telecommunications Conference*, December 2008, pp. 1 – 6.

[5] B. Korte and J. Vygen, *Combinatorial Optimization*. Springer, 2000.

[6] I. Kim, H. Lee, B. Kim, and Y. Lee, "On the Use of Linear Programming for Dynamic Subchannel and Bit Allocation in Multiuser OFDM," in *Proc. of the Global Telecommunications Conference*, November 2001.

[7] R. Urgaonkar and M. Neely, "Opportunistic scheduling with reliability guarantees in cognitive radio networks," in *Proceedings IEEE INFOCOM*, April 2008, pp. 1301 – 1309.

[8] M. Ergen, S. Coleri, and P. Varaiya, "QoS Aware Adaptive Resource Allocation Techniques for Fair Scheduling in OFDMA Based Broadband Wireless Access Systems," *IEEE Trans. Broadcast.*, vol. 49, no. 4, pp. 362–370, 2003.

[9] A. Berger, V. Bonifaci, F. Grandoni, and G. Schäfer, "Budgeted matching and budgeted matroid intersection via the gasoline puzzle," in *IPCO*, 2008, pp. 273–287.

[10] N. Megiddo, "Combinatorial optimization with rational objective functions," *Mathematics of Oper. Res.*, vol. 4, no. 4, pp. 414–424, 1979.

[11] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[12] A. Fiat and G. J. Woeginger, Eds., *Online Algorithms: The State of the Art*, ser. LNCS. Springer, 1998, vol. 1442.

[13] 3GPP, "Evolved Universal Terrestrial Radio Access (EUTRA) Radio Frequency (RF) system scenarios," Technical Specification Group Radio Access Network, TR 36.942, Dec. 2008.

[14] E. Dahlmann, S. Parkvall, J. Sköld, and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband*. Academic Press, 2008.

[15] R. Lougee-Heimer, "The common optimization interface for operations research: Promoting open-source software in the operations research community." *IBM Journal of Research & Development*, vol. 47, no. 1, pp. 57 – 66, 2003.

[16] J. Gross, H. Geerdes, H. Karl, and A. Wolisz, "Performance analysis of dynamic OFDMA systems with inband signaling," *IEEE J. Select. Areas Commun.*, vol. 24, no. 3, pp. 427–436, 2006.

[17] C. H. Papadimitriou and M. Yannakakis, "The complexity of restricted spanning tree problems," *J. ACM*, vol. 29, no. 2, pp. 285–309, 1982.

[18] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, "An optimal algorithm for on-line bipartite matching," in *STOC '90*. New York, NY, USA: ACM, 1990, pp. 352–358.

[19] T. Henttonen, K. Aschan, J. Puttonen, N. Kolehmainen, P. Kela, M. Moisio, and J. Ojala, "Performance of voip with mobility in utra long term evolution," in *Proceedings IEEE VTC*, 2008, pp. 2492 – 2496.