

Extensions of bipartite matchings

Szilvia Keszthelyi

Mathematics BSc

TDK

Supervisor:

Kristóf Bérczi

*ELTE Institute of Mathematics,
Department of Operations Research*

Péter Madarasi

*ELTE Institute of Mathematics,
Department of Operations Research*



Eötvös Loránd University

Faculty of Science

Budapest, 2019

Contents

Introduction	4
1 The landscape of problems	6
1.1 Input models and arrival orders	6
1.2 Online bipartite matching	7
1.3 Competitive ratio	7
1.4 Online vertex-weighted bipartite matching	9
1.5 Adwords	10
1.6 Display Ads	11
1.7 Online submodular welfare maximization	12
1.8 Matroid online bipartite matching	13
1.9 Online submodular maximization under matroid constraint . .	14
1.10 Offline versions	14
2 Online bipartite matching	16
2.1 Algorithms for online bipartite matching	16
2.2 Ranking	17
3 Matroid bipartite matching	23
3.1 H -independent matchings	24
3.1.1 Algorithms	25
3.2 Strongly independent matchings	28
3.2.1 Algorithm for the online problem	28

3.2.2	Bounds for online strongly independent matching . . .	30
-------	---	----

Abstract

Introduction

The purpose of this work is to summarize the results of bipartite matching problems, focusing on the online input models and study their generalizations with matroid constraints.

The well-known theorems of König and Hall characterize the maximum cardinality of a matching in a bipartite graph. In the past decades, numerous extensions of bipartite matching problem have been studied, many of them are motivated by realistic situations. Such model was introduced by Karp, Vazirani, and Vazirani, in online bipartite matching, one side of the bipartition is known in advance and the vertices of the other side arrive in an online fashion. Upon the arrival of a vertex, the incident edges are revealed and an immediate and irrevocable matching decision should be made, possibly leaving the vertex unmatched. The goal is to maximize the size of the obtained matching. In an Internet advertising analogue, one side is the set of items and the other side is the set of buyers. Upon the arrival of a buyer, his preferred items are revealed and a matching should be made (if an available item is left). Furthermore, maximizing the cardinality of a matching with restrictions on the bipartite graph such as capacities on the vertices or matroid constraints are also motivated problems.

Since the publication of the initial paper, online bipartite matching has received considerable attention, therefore Chapter 1 gives a perspective about the allocation problems and the related input models. Although both the offline and online versions of the problems are described, the emphasis is on the latter one.

In Chapter 2, the results in online bipartite matching problem are described including theorems and conjectures of Karp, Vazirani and Vazirani, introduced in 1990 and the latest work of Feige.

Lastly, in Chapter 3, two matroid generalizations of bipartite matching are considered. After a brief overview of the offline problems, greedy algorithms, as possible optimal online algorithms, are studied with the aim of giving bounds for the online problems.

Chapter 1

The landscape of problems

There are numerous generalizations of bipartite matching which are studied in different input models. The previous results achieved regarding the main *online* problems are described in this chapter.

1.1 Input models and arrival orders

There are several input models for bipartite allocation problems. These models differ from each other in the information that the algorithm knows about the arriving vertices.

In the *offline model*, the algorithm knows the whole graph $G(S, T; E)$ ahead of time. In the *online* cases, in contrast, one side of the vertices are known in advance, but vertices of the other side and the edges arrive in an online fashion.

Online problems are studied under *adversarial order* in most of the cases, since it is the most general setting. In this model, no knowledge of the arriving vertices and edges is assumed. The algorithm knows only the vertices of S that have already arrived, edges incident to them and vertices of T . In the case of randomized algorithms, suppose that the adversary knows the code of the algorithm, it can create the worst possible input, but does not know the random choices.

In the *random arrival model*, the adversary selects the graph, but not the arrival order, the vertices of S arrive in a random order.

1.2 Online bipartite matching

In online bipartite matching, there is a bipartite graph $G(S, T; E)$ where one side T , called offline vertices, is known in advance and the other side S arrives online, one vertex at a time. When a vertex $s \in S$ arrives, edges incident to s are revealed. The arriving vertex can be matched to some available neighbor (if any). The goal is to maximize the size of the matching. This problem was introduced by Karp, Vazirani and Vazirani in 1990 [14].

Karp, Vazirani and Vazirani gave a randomized greedy algorithm, called Ranking (see Algorithm 3). An algorithm for the online bipartite matching is called *greedy* if the only vertices of S that it leaves unmatched are those that upon their arrival do not have an eligible neighbor. Ranking chooses uniformly random a permutation π over the vertices of T , π is not known by the adversary, and when an online vertex s arrives, it matches s to the eligible neighbor t with the smallest rank $\pi(t)$ or leaves it unmatched if all the neighbors of s are unavailable. For bipartite graphs with maximum matching of size n , the *competitive ratio* of Ranking is at least $1 - (1 - \frac{1}{n+1})^n$, which tends to $1 - \frac{1}{e}$ as n goes to infinity. In 2018, Feige [9] showed that when the input is chosen from a certain distribution D_n , no online algorithm matches more than $(1 - \frac{1}{e})n + 1 - \frac{2}{e} + \mathcal{O}(\frac{1}{n!})$ edges in expectation thus the bound is essentially tight.

1.3 Competitive ratio

The performance of an online matching algorithm A is quantified by the competitive ratio of A , defined as the ratio of the expected size of matching obtained by A and the size of the maximum matching. The competitive ratio is determined by graphs that have a perfect matching [3, 8]. Hence,

for every n , consider the following class $\mathcal{G}_n$ of bipartite graphs. For every $G(S, T; E) \in \mathcal{G}_n$, $|S| = |T| = n$ and G has a perfect matching. The following notations extend those introduced by Feige [9] and enable the generalization of the competitive ratio which is an algorithm-independent bound. Let G be a bipartite graph $(S, T; E)$, consider the following notations.

- $\rho(A, G)$ is the expected cardinality of matching produced by A when the input graph is G
- $\rho(\mathcal{A}, \mathcal{G})$ is the expected cardinality when A is chosen from a class of algorithms $\mathcal{A}$ and graph $G \in \mathcal{G}$ and $\rho(\mathcal{A}, \mathcal{G}) = \max_{A \in \mathcal{A}} \min_{G \in \mathcal{G}} \rho(A, G)$
- $\rho(A, -) = \rho(A)$ is the minimum over all G of $\rho(A, G)$
- $\hat{\rho}(A) = \min_{G(S, T; E)} \frac{\rho(A, G)}{m}$ where m is the size of the maximum matching i.e. $\hat{\rho}(A)$ is the competitive ratio of A
- $\rho(-, \mathcal{G}) = \rho(\mathcal{G})$ is the maximum over all A of $\rho(A, \mathcal{G})$
- $\hat{\rho} = \inf_n \frac{\rho(\mathcal{G}_n)}{n}$ is the largest constant such that $\hat{\rho}n \leq \rho(\mathcal{G}_n)$ for all n

Competitive ratio is used for measuring the performance of the generalizations of online bipartite matching as well. In this case $\rho(A, G)$ denotes the value of the objective function attained by the algorithm A , if A is randomized then $\rho(A, G)$ is the expected value, and $OPT(G)$ is the optimal solution of the offline problem [18].

$$\hat{\rho}(A) = \min_{G(S, T; E)} \frac{\rho(A, G)}{OPT(G)}$$

In general, let $\hat{\rho}$ be the largest constant such that $\hat{\rho} \leq \min_G \frac{\rho(G)}{OPT(G)}$ where $\rho(G)$ is the maximum over all A of $\rho(A, G)$, as above. The main online allocation problems and their best known ratio $\hat{\rho}$ are summarized in Table 1.2.

1.4 Online vertex-weighted bipartite matching

The simplest generalization of online bipartite matching is when the offline vertices have weights. This was introduced by Aggarwal in [1].

In this version of bipartite matchings, each vertex $t \in T$ has a non-negative weight w_t , which are known ahead of time. The goal is to maximize the sum of the weights of the matched vertices of T .

In this case, a greedy algorithm means that it matches the arriving vertex to the available neighbor with the highest weight. Since Ranking ignores the weights, it does not achieve the ratio of $1 - \frac{1}{e}$ if the weights are highly skewed. One can construct examples in which the ratio of Ranking goes to 0. However, the aforementioned greedy algorithm does well in this case, but has a competitive ratio of $\frac{1}{2}$ when the weights are equal. Algorithm 1, presented in [1], is based on the above algorithms, it runs greedy on perturbed weights, i.e. on $\tilde{w}_t = w_t(1 - e^{r_t-1})$ weights, where r_t is chosen from $U[0, 1]$. Perturbed greedy is a generalization of Ranking. Instead of choosing a random permutation, it selects a permutation from a distribution that depends on the weights of the vertices such that vertices with higher weights have higher probability of being chosen. In the case of equal weights, the algorithm is the same as Ranking since selecting r_t and choosing the highest $1 - e^{r_t-1}$ is identical to choosing a random permutation and picking the highest ranked vertex according to it. Perturbed greedy has competitive ratio of $1 - \frac{1}{e}$ for any set of weights.

Algorithm 1 PERTURBED GREEDY

- 1: **initialize**
 - 2: For every $t \in T$:
 - 3: Pick r_t IID from $U[0, 1]$.
 - 4: Define its perturbed weight as $\tilde{w}_t := w_t(1 - e^{r_t-1})$.
 - 5: **when a vertex $s \in S$ arrives**
 - 6: If s has no available neighbors, continue.
 - 7: Match s to the available neighbor t with the maximum value of $\tilde{w}_t$.
-

1.5 Adwords

The Adwords problem, introduced by Mehta [19], generalizes the online bipartite matching problem as follows. Each offline vertex $t \in T$ has a budget B_t , and edges $(s, t) \in E$ have bids bid_{st} . When a vertex $s \in S$ arrives, it may be matched to a neighbor t that has not spent all its budget. If s is matched to t , t depletes bid_{st} amount of its budget. Once a vertex depletes its entire budget, it becomes unavailable. The goal is to maximize the total budget spent.

The motivation for the Adwords problem comes directly from online advertising [18]. The ad platform, which is the search engine in the case of sponsored search, is in contract with a set of advertisers T , also called bidders. Each advertiser puts in a set of bid values for different keywords they would like to show their ad against. The bidders can also put in a daily budget value B_t representing the maximum they can pay in a day. The search engine gets a sequence of online ad-request $s \in S$, also called queries and ad-slots, as users search during the day. When a request s arrives, the search engine sees the $bid_{s,t}$ for each bidder $t \in T$. Each ad-request can be allocated to one advertiser and the winning advertiser is charged its bid. Once an advertiser run out of its budget, it cannot be allocated to any more ad-slots. The objective in this model is to maximize the total amount of money spent by the advertisers. In some model, if an advertiser's remaining budget B'_t is less than its bid $bid_{s,t}$, then the search engine runs with B'_t as a new bid.

Small bid assumption

Adwords problem with small bid assumption is a special case of the problem above, motivated by realistic situations. For each edge (s, t) bid_{st} is small compered to B_t . Since, remaining at the previous example, advertisers typically want a large number of ad clicks per day. The Adwords problem is mostly studied under the small bid assumption. When the bids are small,

the algorithms can ignore the above correction of the bids, since overspending is not significant.

Algorithms

The best known competitive ratio differs between algorithm for Adwords with general bids and with small bids. *Greedy*, which allocates each s to that available t which has the maximum value of bid_{st} , achieves a ratio of $\frac{1}{2}$. The following method is $(1 - \frac{1}{e})$ -competitive, but only works in the case of small bids. *MSSV* scales the bids with a function similar to the one used by Perturbed greedy, and then runs greedy on the scaled bids.

Adwords problem generalizes both online bipartite matching and online vertex-weighted matching problems. The special case in which $B_t = w_t$ for all t and $bid_{st} = w_t$ for all $(s, t) \in E$, corresponds with the vertex-weighted version and online bipartite matching is equivalent with Adwords when all budgets and bids are equal to 1. However, neither problem is a special case under the small-bids assumption.

1.6 Display Ads

Display Ads problem is an edge-weighted and capacitated generalization of online bipartite matching motivated by the display ads applications [10, 18]. In this problem, each edge $(s, t) \in E$ has a weight w_{st} and each vertex $t \in T$ has a capacity c_t which is an upper bound on the number of vertices that can be matched to t . When a vertex $s \in S$ arrives, it may be assigned to an available neighbor t . The goal is to maximize the total weight of the selected edges.

Practical examples fulfill the so-called *large capacity assumption*, that is the capacities of the vertices are significantly larger than the weights of the edges. This special case was considered in [10, 18].

Relevant input models

If there is only one offline vertex and its capacity is $c_t = 1$, then the problem is identical to picking the maximum from a stream of numbers which arrive online therefore it is not possible to obtain any non-trivial competitive ratio for this model, since as soon as an algorithm chooses a number, the adversary can generate a much bigger number subsequently. The random order input model and the *free-disposal model* are two motivated models, in which a better competitive ratio can be reached. In the free-disposal model, edges can be removed later from the matching, but this choice is irrevocable.

1.7 Online submodular welfare maximization

Online submodular welfare maximization is a generalization of all the problems above (see in Table 1.1). Each offline vertex $t \in T$ has a non-negative monotone submodular valuation function $f_t: 2^S \rightarrow \mathbb{R}^+$. When a vertex $s \in S$ arrives, it have to be allocated to a neighbor t . The allocation produces a partition of S , let S_t be the vertices of S allocated to t . The goal is to maximize $\sum_{t \in T} f_t(S_t)$.

A set function $f: 2^S \rightarrow \mathbb{R}^+$ is called *monotone* if $f(Y) \leq f(X)$ for every two sets $Y \subseteq X \subseteq S$, and *submodular* if for any two sets X and Y , $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$. An equivalent form to submodularity is $f(Y \cup x) - f(Y) \geq f(X \cup x) - f(X)$ for every two sets $Y \subseteq X \subseteq S$ and an element $x \in S \setminus X$.

Algorithm

For online submodular welfare maximization problem, algorithm 2, which is a greedy method, achieves a competitive ratio of $\frac{1}{2}$ [18]. Kapralov showed in [13] that this ratio is optimal when vertices of S arrive in an adversarial order and the allocations are irrevocable. However, in the setting when vertices arrive in a uniformly random order, Buchbinder [5] proved that the same algorithm

achieves a strictly greater competitive ratio. For submodular maximization, a greedy algorithm means that it allocates each $s \in S$ to $t \in T$ with the highest marginal gain.

Algorithm 2 GREEDY FOR ONLINE SUBMODULAR MAXIMIZATION

- 1: **when a vertex** $s \in S$ **arrives**
 - 2: For each $t \in T$, let S_t denote the set of vertices in S already allocated to t .
 - 3: Allocate s to that t which maximizes $f_t(S_t \cup s) - f_t(S_t)$.
-

Table 1.1: The aforementioned problems are special cases of online submodular welfare maximization with the valuation functions below.

Problem	Valuation function
Bipartite matching	$\min\{1, S_t \}$
Vertex-weighted matching	$w_t \min\{1, S_t \}$
Adwords	$\min\{B_t, \sum_{s \in S_t} bid_{st}\}$
Display ads	$\max_{X \subseteq S_t, X \leq c_t} \{\sum_{s \in X} w_{st}\}$

1.8 Matroid online bipartite matching

The problems above can be generalized with matroid constraints. See definitions related to matroids in Chapter 3. A version of these generalizations is when the objective is to maximize the size of matching while the set of matched offline vertices are independent in a given matroid. In the case of online bipartite matching, offline vertices T and a matroid on T are known in advance and vertices of S arrive online, one at a time along with their incident edges. Wang and Wong gave an $1 - \frac{1}{e}$ -competitive algorithm in the random arrival model [23].

Adwords problem can be generalized this way as well and a greedy algorithm presented in [23] achieves a ratio of $1 - \frac{1}{e}$ with the small bid assumption. This is a motivated generalization, since in Adwords problem, each bidders have separate budgets, but many advertisers, for example who are selling several products, are typically hosting multiple ads, but only be able to budget for each ad independently. If an advertiser manages multiple ads with a fixed total budget, it corresponds to the matroid generalization of Adwords, with a partition matroid on T .

1.9 Online submodular maximization under matroid constraint

A matroid generalization of submodular maximization was studied in [7]. Each offline vertex $t \in T$ has a non-negative monotone submodular function $f_t: 2^S \rightarrow \mathbb{R}^+$ and a matroid $H = (S, \mathcal{I})$ is given on S . As a vertex $s \in S$ arrives, it may be assigned to a vertex $t \in T$ with respect to the matroid constraint, that is the vertices allocated to t have to form an independent set $S_t \in \mathcal{I}$ for each $t \in T$. The objective is to maximize $\sum_{t \in T} f_t(S_t)$, just as in Section 1.7. Online bipartite matching is a special case under a 1-uniform matroid constraint.

In the *free-disposal model*, one may remove some of the matched vertices in order to ensure the independence of set S_t . In [7], authors gave a $\frac{1}{1+\alpha}$ -competitive algorithm with free-disposal, where α is the root of $\alpha = e^{\alpha-2}$.

1.10 Offline versions

In the offline problem, the entire graph is known in advance. The well-known theorems of Hall and König give conditions on the existence of a matching covering one side of the bipartition and on the maximum cardinality of a matching in a bipartite graph [11, 17]. The offline version of vertex-weighted

bipartite matching and display ads problems can be solved in polynomial time as well. However, the Adwords problem (without the small bid assumption) is NP-hard to approximate to any factor better than $\frac{15}{16}$ and the best known offline algorithm is $\frac{3}{4}$ approximation [6, 18, 22]. Adwords problem with small bid assumption can be solved with a $(1 - 4\varepsilon)$ approximation if the bid to budget ratio $\frac{bid_{st}}{B_t}$ is at most ε for all s, t [6, 18]. The Submodular Welfare Maximization problem is NP-hard to approximate to better than $1 - \frac{1}{e}$ [15, 18]. Rado's theorem [21] generalizes Hall's theorem for the offline matroid bipartite matching problem, which is detailed in Section 3.1.

Table 1.2: Summary of result of the best known ratios for online matching problems. Question marks correspond to open questions.

Problem	$\hat{\rho}$
Bipartite matching	$1 - \frac{1}{e}$
Vertex-weighted matching	$1 - \frac{1}{e}$
Adwords	$\frac{1}{2}$ (?)
• with small bids	$1 - \frac{1}{e}$
Display ads	0
• in random order	$\frac{1}{e}$
• with free disposal	$\frac{1}{2}$ (?)
• with free disposal and large bids	$1 - \frac{1}{e}$ (?)
Submodular welfare maximization	$\frac{1}{2}$
Matroid bipartite matching (in random order)	$1 - \frac{1}{e}$

Chapter 2

Online bipartite matching

As above, in online bipartite matching problem, the input is a bipartite graph $G(S, T; E)$ where the offline vertices T are known in advance and the vertices of S arrive online. Edges incident to $s \in S$ are revealed when s arrives and s may be matched to an available adjacent vertex $t \in T$. The goal is to maximize the size of the matching.

2.1 Algorithms for online bipartite matching

When a vertex s arrives, an algorithm may match s to one of its previously unmatched neighbors or leave it unmatched and this choice is irrevocable.

An algorithm for the online bipartite matching is called *greedy* if the only vertices of S that it leaves unmatched are those that upon their arrival do not have an eligible neighbor. Every non-greedy algorithm A can be replaced by a greedy method A' that produces at least as many edges as A does since for any vertex s that A left unmatched even though it had available neighbor at the arriving, there are two cases. In the first case, when at the end of the algorithm s still has an unmatched neighbor t , the matching produced by A can be extended with edge (s, t) to get a better solution. If a greedy A' matches s to t and does the same as A otherwise, it has strictly better performance. In the other case, when A matches all the neighbors of s , let

greedy A' differ from A in matching s to a neighbor t , which is available at the arrival of s . Thus, all the vertices of T that are matched by A are matched by A' as well, hence A' matches at least as many vertices as A does.

Every greedy algorithm produces a maximal matching and a maximal matching is at least half of the maximum matching, since an edge can exclude at most two edges from the maximum matching.

For every deterministic algorithm, the adversary can select an input on which the algorithm produces only half the size of the maximum matching. For example, let the first $\frac{|S|}{2}$ arriving vertices be incident to all the vertices of T and let the remaining vertices of S be neighbors only of vertices that the algorithm matches to the first $\frac{|S|}{2}$ vertices.

2.2 Ranking

To improve the size of the matching beyond $\frac{n}{2}$, Karp, Vazirani and Vazirani [14] considered randomized methods. Algorithm 3 is the randomized greedy method of Karp, Vazirani and Vazirani [14], called Ranking. For bipartite graphs with maximum matching of size n , Ranking matches at least $(1 - \frac{1}{e})n$ edges in expectation. They showed that this bound is essentially tight. When the input is chosen from a certain distribution $\mathcal{D}_n$, no online algorithm matches more than $(1 - \frac{1}{e})n + \mathcal{O}(1)$ edges in expectation. In 2018, Feige [9] gave a tighter bound, $(1 - \frac{1}{e})n + 1 - \frac{2}{e} + \mathcal{O}(\frac{1}{n!})$ serves as an upper bound on the performance ratio of any online matching algorithm.

Algorithm 3 RANKING

- 1: **initialize**
 - 2: Choose a random permutation π of T .
 - 3: **when a vertex $s \in S$ arrives**
 - 4: If s has no available neighbors, continue.
 - 5: Match s to the neighbor t with the smallest value of $\pi(t)$.
-

The aforementioned distribution $\mathcal{D}_n$ is defined as follows. For each graph from $\mathcal{D}_n$, let the cardinality of the two sides be n i.e. $|S| = |T| = n$. Select a permutation τ over T . For every $i = 1 \dots n$, $(s_j, t_i) \in E$ iff $j = \tau(k)$ for some $k \geq i$. $\mathcal{D}_n$ satisfies the following three properties which support the conjecture that a graph from $\mathcal{D}_n$ is the worst input against Ranking.

- It has a unique perfect matching, which is the set of edges $(s_j, t_{\tau(j)})$ for $1 \leq j \leq n$.
- If s_j cannot be matched, neither can any of the vertices afterwards since all neighbors of $s_j + 1$ are also neighbors of s_j . Hence all the vertices in a prefix of S are matched and then no vertices in the resulting suffix are matched.
- An offline vertex $t_{\tau(j)}$ can be matched only in the first j rounds.

The expected size of the matching produced by Ranking is the same on every graph from $\mathcal{D}_n$ since Ranking generates a random permutation over T , hence graphs that only differ by a permutation over T considered equal. Such as the graphs of $\mathcal{D}_n$. Thus, consider one representative graph, called MonotoneG [9], which is shown on Figure 2.1. Permutation τ of MonotoneG is the identity permutation i.e. the set of the edges are $E = \{(s_i, t_j) | 1 \leq i \leq n, i \leq j \leq n\}$, so the perfect matching is $M = \{(s_i, t_i) | 1 \leq i \leq n\}$.

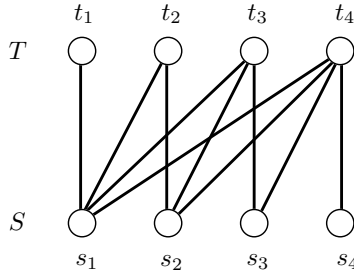


Figure 2.1: A representative graph of $\mathcal{D}_4$.

It is known that Ranking is optimal against $\mathcal{D}_n$, but it is an open question whether $\mathcal{D}_n$ is optimal against Ranking.

Conjecture 2.2.1. $\rho(\mathcal{G}_n) = \rho(\text{Ranking}, \mathcal{D}_n)$ for all n .

The above conjecture of Karp, Vazirani and Vazirani is still open, but the bounds are essentially tight. The theorems below summarize the known bounds on $\hat{\rho}$. Firstly, Mehta [19] gave an upper bound.

Theorem 2.2.2. For $\mathcal{D}_n$ defined as above, $\rho(\mathcal{D}_n) \leq (1 - \frac{1}{e})n + \mathcal{O}(1)$.

Feige [9] proved that exists a tighter bound on $\rho(\mathcal{D}_n)$.

Theorem 2.2.3. Let the function $a(n)$ be such that $\rho(\text{Ranking}, \mathcal{D}_n) = \frac{a(n)}{n!}$ for all n . Then $a(n) = (n+1)! - d(n+1) - d(n)$, where $d(n)$ is the number of derangements (permutations with no fixed points) on the numbers $1, \dots, n$. Consequently, $\rho(\mathcal{D}_n) = (1 - \frac{1}{e})n + 1 - \frac{2}{e} + \mathcal{O}(\frac{1}{n!})$, and this is an upper bound on $\rho(\mathcal{G}_n)$.

Karp, Vazirani and Vazirani [14] showed a lower bound on $\rho(\mathcal{G}_n)$, but the presented proof had gaps. Later, Goel and Mehta [12, 19] gave alternative proofs motivated by the Adwords problem.

Theorem 2.2.4. For each bipartite graph $G \in \mathcal{G}_n$, the competitive ratio of Ranking algorithm is at least $1 - (1 - \frac{1}{n+1})^n$, which tends to $1 - \frac{1}{e}$ as n goes to infinity. Hence $\rho(\text{Ranking}) \geq (1 - \frac{1}{e})n$ and $\hat{\rho} \geq 1 - \frac{1}{e}$.

Proof of Theorem 2.2.4

The proof below is based on the one presented by Birnbaum and Mathieu in [3]. Let $\text{Ranking}(G, \sigma, \pi)$ denote algorithm Ranking on input graph G when the arrival order is permutation σ over S and the ranking is π over T . When G and σ are fixed, $\text{Ranking}(\pi)$ is the simplification of notation $\text{Ranking}(G, \sigma, \pi)$. Let $M_{\sigma, \pi}$ and M_π be the matchings produced by $\text{Ranking}(G, \sigma, \pi)$ and $\text{Ranking}(\pi)$ respectively. Let $M(v)$ be the neighbor of vertex u in a given matching M .

Claim 2.2.5. The competitive ratio is determined by graphs that have a perfect matching.

Proof. Let G be an arbitrary bipartite graph and fix σ and π . Let v be a vertex, $G' = G \setminus v$ and σ' and π' be the permutations over S' and T' induced by σ and π respectively. If the matchings $M_{\sigma,\pi}$ and $M_{\sigma',\pi'}$ are not identical, then they differ by a single alternating path starting at vertex v , see in Lemma 2.2.6. Thus, for every σ and π , $M_{\sigma,\pi}$ has at least as many edges as $M_{\sigma',\pi'}$. For all σ and π , $|M_{\tilde{\sigma},\tilde{\pi}}| \leq |M_{\sigma,\pi}|$ when $\tilde{G} \in \mathcal{G}_n$ for some n is the graph constructed from G by removing all vertices which are not in the maximum matching, $\tilde{\sigma}$ and $\tilde{\pi}$ are the permutations over $\tilde{S}$ and $\tilde{T}$. Therefore the size of the maximum matching is the same in both graphs, hence the competitive ratio does not increase. $\square$

Lemma 2.2.6. *Let $v \in S \cup T$, $G' = G \setminus v$, and $\text{Ranking}(G', \sigma', \pi')$ is defined as above. If the matchings produced by $\text{Ranking}(G, \sigma, \pi)$ and $\text{Ranking}(G', \sigma', \pi')$ are not identical, then they differ by a single alternating path starting at vertex v .*

Proof. In set $M_{\sigma,\pi} \cup M_{\sigma',\pi'}$, the degree of each vertex is at most 2, therefore it is a disjoint union of paths and circles. If $M_{\sigma,\pi} \cup M_{\sigma',\pi'}$ is a matching, then the two matchings are identical or $M_{\sigma,\pi}(v)$ cannot be matched after removing v . It is needed that if there is a path of length at least 2 than it is unique and starts at vertex v .

Suppose that there is a path P of length k where $k > 1$, which does not include v . Consider the first arriving $s \in S$ of P . If s is matched to t in one case, but for the other input graph, (s, t) is not in the matching, then in the second case, t must be matched to an earlier vertex which is a contradiction. Thus such path does not exist. And circles are not contained by $M_{\sigma,\pi} \cup M_{\sigma',\pi'}$ since a circle means that Ranking matches the same vertices in different order in the two cases which is contradict to the construction of σ' and π' . $\square$

Hence assume that G has a perfect matching. Fix graph G and arrival order σ . Let M^* be a perfect matching, and $n = |S|$. The following two Lemmas are structural observations focusing on the ranks of the matched vertices.

Lemma 2.2.7. *Fix $s \in S$ and let $t = M^*(s)$. If t is not matched by $\text{Ranking}(\pi)$, then s is matched to a vertex t' whose rank $\pi(t')$ is less than $\pi(t)$.*

Proof. If t is not matched by $\text{Ranking}(\pi)$, then when s arrives, it has some eligible neighbors since t is one. Thus, s is matched to the eligible neighbor t' with the minimum rank. $\square$

Lemma 2.2.8. *Let $s \in S$ and $t = M^*(s)$. Let π' be a permutation and let π_i be the permutation obtained from π' by changing the rank of vertex t to i in other words, removing t from π' and putting it back in so its rank is i . If t is not matched by $\text{Ranking}(\pi')$, then for every i , s is matched by $\text{Ranking}(\pi_i)$ to a vertex t_i with rank $\pi_i(t_i)$ is at most $\pi'(t)$.*

Proof. According to Lemma 2.2.7, s is matched to a vertex t' in $M_{\pi'}$ and $\pi'(t') < \pi'(t)$. As in Lemma 2.2.6, if matchings $M_{\pi'}$ and M_{π_i} are not identical, then they differ along a single alternating path starting at vertex t with an edge of M_{π_i} since t is unmatched by $\text{Ranking}(\pi')$. Moreover, the path is monotone, i.e. the vertices of T traversed have increasing rank in π_i . Thus, s is matched to vertex t_i by $\text{Ranking}(\pi_i)$, whose rank is $\pi_i(t_i) \leq \pi_i(t')$. By definition of π_i , $\pi_i(t') \leq \pi'(t') + 1 \leq \pi'(t)$. $\square$

Lemma 2.2.9. *Let x_k denote the probability over π that the vertex of T of rank k is matched by the algorithm. Then $1 - x_k \leq \frac{1}{n} \sum_{1 \leq j \leq k} x_j$.*

Proof. Let $R_k \subseteq S$ denote the set of vertices that is matched by the algorithm to the vertices of T of rank less than or equal to k . Given a random permutation π over T , define a new random permutation π' obtained by choosing a vertex $t \in T$ uniformly at random, taking it out of π and moving it back in so that its rank is k . Consider the matching produced by $\text{Ranking}(\pi')$. Let s be such that $t = M^*(s)$. The probability that t is not matched by $\text{Ranking}(\pi')$ is $1 - x_k$. By Lemma 2.2.8, applied for i such that $\pi_i = \pi$, if t is not matched by $\text{Ranking}(\pi')$, then s is matched by $\text{Ranking}(\pi)$ to a vertex $\tilde{t}$ such that $\pi(\tilde{t}) \leq k$, in other words, $s \in R_k$. Choosing t is random, so s

as well and Lemma 2.2.8 holds for all i , thus s is independent of π , hence independent of R_k . In other words, the event that $s \in R_k$ does not depend of π . Thus, conditional on π , the event that $s \in R_k$ has probability $\frac{|R_k|}{n}$. Taking expectations over π , this equals to $\frac{1}{n} \sum_{1 \leq j \leq k} x_j$. $\square$

With Lemma 2.2.9, the proof of Theorem 2.2.4 can be completed. Since the considered graphs have perfect matchings, the competitive ratio is the infimum of $\frac{1}{n} \sum_{1 \leq j \leq n} x_j$. Let $s_k = \sum_{1 \leq j \leq k} x_j$. Rewrite x_k as $s_k - s_{k-1}$ to get an equivalent form of Lemma 2.2.9 which is $1 + s_{k-1} \leq s_k(1 + \frac{1}{n})$. The infimum occurs when all inequalities are tight equalities. This yields a recursive formula $s_k = \frac{n}{n+1} + \frac{ns_{k-1}}{n+1}$. It follows that $s_k = \sum_{1 \leq j \leq k} (1 - \frac{1}{n+1})^j$. Hence, the competitive ratio is at least $\frac{1}{n} s_n = \frac{1}{n} \sum_{1 \leq j \leq n} (1 - \frac{1}{n+1})^j = 1 - (1 - \frac{1}{n+1})^n$ which tends to $1 - \frac{1}{e}$ as n goes to infinity.

Chapter 3

Matroid bipartite matching

In this section, two matroid bipartite matching problems are studied. The first one is when there is a matroid on one side of the bipartition, in the online problem, this side is the offline vertices T . This online version and related results were briefly mentioned in Section 1.8. The second one, when there is a matroid on the other side as well, has more open questions in the online case.

Matroids were introduced by Whitney [24] and independently by Nakasawa [20, 2] as abstract generalizations of linear independence in vector spaces. A matroid H is a pair $(S, \mathcal{I})$ where S is the *ground set* of the matroid and $\mathcal{I} \subseteq 2^S$ is a family of subsets of S that satisfy the following, so-called independence axioms.

$$(I1) \quad \emptyset \in \mathcal{I},$$

$$(I2) \quad X \subseteq Y \in \mathcal{I} \Rightarrow X \in \mathcal{I},$$

$$(I3) \quad \text{For every subset } X \subseteq S, \text{ the maximal subsets of } X \text{ which are in } \mathcal{I} \text{ have the same cardinality.}$$

The members of $\mathcal{I}$ are called *independent* sets, all other subsets of S are *dependent*. Axiom (I3) means that the maximal independent subsets of each subset of S are equicardinal. This maximum number is called the *rank* of X

and is denoted by $r(X)$, where r is the *rank function* of the matroid and $r(S)$ is the *rank of the matroid*. For a set S , the matroid in which every subset of S is independent is called a *free matroid*. As other examples to matroids, uniform and partition matroids are defined as follows. Let $\{S_1, \dots, S_k\}$ be a partition of S and let $g_1, \dots, g_k$ be non-negative integers. Declare a subset $I \subseteq S$ independent if $|I \cap S_i| \leq g_i$ holds for each i . The matroid obtained in this way is called a *partition matroid*. Its rank function r is given by $r(X) = \sum_i (\min\{g_i, |X \cap S_i|\})$. The *uniform matroid* is a partition matroid in the special case of $k = 1$ [11].

3.1 H -independent matchings

Let $G = (S, T; E)$ be a simple bipartite graph and H a matroid on T . A matching is called *H -independent*, if the vertices of T covered by the matching, is independent in H . Note that H_E -independence is also used, since H determines a matroid on E in which a subset F is independent if $d_F(t) \leq 1$ for every $t \in T$ and the subset of T covered by F is independent in H . Rado [21] gave a matroid generalization of Hall's Theorem. For a subset $X \subseteq S$, let $\Gamma(X)$ denote the set of neighbors, i.e. $\Gamma(X) = \{t \in T : \text{there is an edge } (s, t) \in E \text{ with } s \in X\}$ [11].

Theorem 3.1.1 (Rado). *Let $G = (S, T; E)$ be a bipartite graph and let H be a matroid on T . G has an H -independent matching covering S if and only if the following Rado condition holds,*

$$r(\Gamma(X)) \geq |X|$$

for every subset $X \subseteq S$.

The defect form of Rado's theorem gives the maximum cardinality of the H -independent matching when the entire S cannot be covered and it depends only on the extent of the violation of the previous condition.

Theorem 3.1.2 (Rado, defect form). *Let $G = (S, T; E)$ be a bipartite graph and H a matroid on T . The maximum cardinality of an M -independent matching is equal to*

$$\mu := \min_{X \subseteq S} \{r(\Gamma(X)) + |S - X|\}.$$

As the above theorems show, the conditions for the offline optimum are known.

In the online version of matroid bipartite matching, offline vertices T and a matroid H on T are known in advance and vertices of S arrive online, one at a time along with their incident edges, just as in Section 1.8. The objective is to maximize the size of an H -independent matching.

3.1.1 Algorithms

In general, a *greedy* algorithm for matroid online bipartite matching is similar to the one in Section 2.1 for online bipartite matching, it only leaves $s \in S$ unmatched if upon its arrival all of its neighbors are already matched or it has no unmatched neighbor t such that s and t are both form an independent with the previously matched vertices of S and T respectively in the appropriate matroid. In this problem, the only matroid is on T , therefore consider the above definition with the free matroid on S .

When the matroid is on T , then greedy is optimal, since for every non-greedy algorithm A there exists a greedy A' that produces at least as many edges as A does, as follows. In the case of matroid online bipartite matching, there are three instances when a vertex s is left unmatched by A even though it had available neighbor at the arriving. Consider the matching produced at the end of algorithm A . If s has a neighbor t such that the set of the matched vertices of T and t is independent in the matroid, then the argument in Section 2.1 can be applied. It occurs when t is matched by A to a latter $s' \in S$ or it is one case when there is an unmatched neighbor t . The other one is when t is unmatched but the matched vertices and t are dependent all together, then there exists a circuit which contains t and some latter vertices

and from which any vertex can be removed, so that the remaining set is independent. Thus, a vertex t' which is matched to a vertex s' that arrived later than s can be left out. So, if greedy A' matches s to t and leaves out edge (s', t') , otherwise runs as A , then A' matching at least as many vertices as A does.

Definition 3.1.3. An algorithm is a *factor α approximation* for a problem iff for every instance of the problem it can find a solution within a factor α of the optimum solution.

If the problem at hand is a minimization then $\alpha > 1$ and this definition implies that the solution found by the algorithm is at most α times the optimum solution. If the problem is a maximization, $\alpha < 1$ and this definition guarantees that the approximate solution is at least α times the optimum [16].

Claim 3.1.4. *Greedy is $\frac{1}{2}$ approximation algorithm for the H -independent matching problem.*

Proof. Let $G = (S, T, E)$ be a bipartite graph and H a matroid on T . Consider the following two matchings of G , the maximum H -independent matching produced by the greedy algorithm and a fixed optimal H -independent matching. Let P denote the vertices of T which are matched by the greedy algorithm and let R be those vertices of T which are covered by the optimal matching. Furthermore, let $P_1 = P \cap R$, $P_2 = P \setminus P_1$, $R_2 = R \setminus (R_1 \cup P_1)$ and $t \in R_1$ iff t has a neighbor s such that edge (s, t) is in the maximal matching and s is matched by the greedy algorithm too, see in figure 3.1.

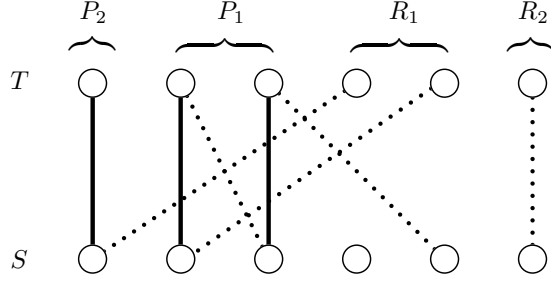


Figure 3.1: Subsets of T . Dotted edges corresponds to the optimal matching and the solid ones are produced by the greedy algorithm.

The claim is equivalent with the following inequality.

$$|R| \leq 2|P|$$

It can be written in a simplified form by using the above defined subsets.

$$\begin{aligned} |P_1| + |R_1| + |R_2| &\leq 2(|P_1| + |P_2|) \\ |R_1| + |R_2| &\leq |P_1| + 2|P_2| \end{aligned}$$

Decompose it into the sum of two inequalities.

$$|R_1| \leq |P_1| + |P_2| \tag{3.1}$$

$$|R_2| \leq |P_2| \tag{3.2}$$

Inequality (3.1) holds by the definition of R_1 . Note some structural observation for inequality (3.2). Sets $P_1 \cup P_2$ and $P_1 \cup R_2$ are independent, since both are covered by one of the H -independent matchings, hence $r(P_1 \cup P_2) = |P_1| + |P_2|$ and $r(P_1 \cup R_2) = |P_1| + |R_2|$. For all $t \in R_2$, $P_1 \cup P_2 \cup \{t\}$ is dependent, since by the definition of R_2 , when some $s \in S$ arrived, it had available neighbor t , yet unmatched. It occurs only when t is dependent with the already matched vertices, since the algorithm is greedy. Thus $r(P_1 \cup P_2 \cup R_2) = r(P_1 \cup P_2)$. However, $P_1 \cup R_2 \cup \{p\}$ can be independent for some $p \in P_2$, hence $r(P_1 \cup R_2) \leq r(P_1 \cup R_2 \cup \{p\}) \leq r(P_1 \cup R_2 \cup P_2)$. By

putting these together, inequality (3.2) follows.

$$\begin{aligned} |P_1| + |R_2| &= r(P_1 \cup R_2) \leq r(P_1 \cup R_2 \cup P_2) = r(P_1 \cup P_2) = |P_1| + |P_2| \\ |R_2| &\leq |P_2| \end{aligned}$$

□

3.2 Strongly independent matchings

Let $G = (S, T; E)$ be a bipartite graph and H_1 and H_2 be matroids on S and T , respectively. A matching is called *strongly independent* if it covers an independent set of vertices in both matroids. Brualdi observed a formulation which includes the theorems of König and Rado as special cases [4, 11].

Theorem 3.2.1 (Brualdi). *Let $G = (S, T; E)$ be a bipartite graph with matroids H_1 and H_2 on ground-sets S and T , respectively. The largest cardinality of a strongly independent matching of G is equal to*

$$\min\{r_1(X) + r_2(Y) : X \subseteq S, Y \subseteq T, X \cup Y \text{ covers each edge of } G\}.$$

3.2.1 Algorithm for the online problem

In the online version, only offline vertices T and matroid H_2 on T are known in advance and vertices of S , along with H_1 arrive online. The objective is to maximize the size of a strongly independent matching.

Greedy, defined above, is optimal when there is no matroid on the online vertices, however, the above arguments cannot be applied in the case of two matroids. In the following instance, one cannot guarantee that the greedy matching has at least as many edges as the non-greedy one. Let s_1 be a vertex which has available neighbor t_1 at the arriving, but the algorithm leaves s_1 unmatched. If latter s_2 is matched to t_1 and s_3 to a t_2 and $\{s_1, s_3\}$ with some previously matched vertices is dependent, then greedy that matches s_1 , may produce strictly less edges, since it is possible that both s_2 and s_3 remain

unmatched. Yet, greedy has the same approximation factor with and without the matroid on S . Note, that Claim 3.1.4 is a special case when H_1 is the free matroid, but different methods were used in the two proofs, hence both are presented.

Claim 3.2.2. *Greedy is $\frac{1}{2}$ approximation algorithm for the strongly independent matching problem.*

Proof. Let $e_1, \dots, e_k$ be the edges produced by the greedy algorithm. Let $\theta_i = \max\{|F| : F \subseteq E, \{e_1, \dots, e_i\} \subseteq F \text{ and } F \text{ is a matching}\}$. The approximation factor follows from the inequality $\theta_i \geq \theta_{i-1} - 1$ for all i .

Let M_i be a matching attaining the minimum in $\{|F| : F \subseteq E, \{e_1, \dots, e_i\} \subseteq F \text{ and } F \text{ is a matching}\}$. Consider the cases of how the size of the matching changes when e_i is added to the edges of M_{i-1} in condition of some edges may be removed. Fix an M_{i-1} . If $e_i \in M_{i-1}$ then $\theta_i = \theta_{i-1}$ and the above inequality holds.

If $e_i = (s_i, t_i) \notin M_{i-1}$, then some conditions on s_i and t_i occurs, namely whether they are covered by another edge from M_{i-1} or form a dependent with some matched vertices. It is needed that by adding e_i to M_{i-1} but losing some other edges due to the conditions, the inequality still holds in all the possible cases, which follows.

Let call a vertex v available if it is not dependent of some of the vertices covered by M_{i-1} nor covered by it. Note, that these cannot occur at the same time and the roles of s_i and t_i are symmetric.

When one of s_i and t_i is available, there exists M'_{i-1} such that $e_i \in M'_{i-1}$ and $|M'_{i-1}| = \theta_{i-1}$, as follows. Both s_i and t_i cannot be available, since then M_{i-1} would not be maximal. If t_i is available and s_i is dependent from a set $S' \subseteq S$ which is covered by M_{i-1} . Then there exists a circuit which contains s_i and from which removing a vertex s' , s_i is independent with the remaining matched vertices, hence e_i can be added with the loss of only one other edge. The other case is when t_i is available, is when s_i is covered by another matched edge. Leaving that edge and adding e_i instead makes no change in θ_{i-1} .

In the following three cases, at most two edges are needed to be removed in order to e_i can be added. Let t_i be in a circuit with some vertices of $T' \subseteq T$, covered by M_{i-1} . If s_i is in a circuit with some vertices of S' , then in the appropriate circuits exists an s' and a t' such that removing them, both s_i and t_i are independent with the remaining matched vertices. This means that the edges, which cover s' and t' , are removed, but e_i is added which makes a loss of at most one edge in total. If s_i is covered by M_{i-1} , then removing the edges which cover t' , defined as above, and s_i is sufficient to add e_i .

As the last instance, let t_i be covered by M_{i-1} , then s_i is also covered by M_{i-1} , but $(s_i, t_i) \notin M_{i-1}$. Replacing the two edges that cover s_i and t_i with e_i makes a strongly independent matching of size $\theta_{i-1} - 1$.

The above cases shows that the matchings derived from M_{i-1} has at least $\theta_{i-1} - 1$ edges and θ_i , by definition, is at least as much as the size of these matchings, hence the inequality holds. By definition, θ_0 is the cardinality of the maximal strongly independent matching and θ_k is the cardinality of the greedy matching. An inequality between θ_0 and θ_k follows from $\theta_i \geq \theta_{i-1} - 1$.

$$\begin{aligned} k = \theta_k &\geq \theta_{k-1} - 1 \geq \dots \geq \theta_0 - k \\ 2k &\geq \theta_0 \end{aligned}$$

In other words, the size of the matching produced by greedy is at least $\frac{1}{2}$ times the optimum. $\square$

3.2.2 Bounds for online strongly independent matching

Claim 3.2.2 gives a lower bound for matroid online bipartite matching, greedy algorithms have a competitive ratio of at least $\frac{1}{2}$, since they always produce a matching of size at least half of the optimum. The results of online bipartite matching are serve as an upper bound, since it is a special case, when the matroids are the free matroid. Hence a matroid online bipartite matching algorithm has a competitive ratio of at most $1 - \frac{1}{e}$. However, the tight bound is an open question.

Bibliography

- [1] G. Aggarwal, G. Goel, C. Karande, and A. Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1253–1264. SIAM, 2011.
- [2] K. Bérczi and T. Swarcz. Complexity of packing common bases in matroids. *arXiv preprint arXiv:1903.03579*, 2019.
- [3] B. E. Birnbaum and C. Mathieu. On-line bipartite matching made simple. *Sigact News*, 39(1):80–87, 2008.
- [4] R. A. Brualdi. Common transversals and strong exchange systems. *Journal of Combinatorial Theory*, 8(3):307–329, 1970.
- [5] N. Buchbinder, M. Feldman, and M. Garg. Online submodular maximization: Beating $1/2$ made simple. *arXiv preprint arXiv:1807.05529*, 2018.
- [6] D. Chakrabarty and G. Goel. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. *SIAM Journal on Computing*, 39(6):2189–2211, 2010.
- [7] T. H. Chan, Z. Huang, S. H.-C. Jiang, N. Kang, and Z. G. Tang. On-line submodular maximization with free disposal: Randomization beats $1/4$ for partition matroids. In *Proceedings of the Twenty-Eighth An-*

- nual ACM-SIAM Symposium on Discrete Algorithms*, pages 1204–1223. SIAM, 2017.
- [8] A. Eden, U. Feige, and M. Feldman. Max-min greedy matching. *arXiv preprint arXiv:1803.05501*, 2018.
 - [9] U. Feige. Tighter bounds for online bipartite matching. *arXiv preprint arXiv:1812.11774*, 2018.
 - [10] J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating $1-1/e$. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009.
 - [11] A. Frank. Connections in combinatorial optimization. *Discrete Applied Mathematics*, 160(12):1875, 2012.
 - [12] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 982–991. Society for Industrial and Applied Mathematics, 2008.
 - [13] M. Kapralov, I. Post, and J. Vondrák. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1216–1225. SIAM, 2013.
 - [14] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.
 - [15] S. Khot, R. J. Lipton, E. Markakis, and A. Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. *Algorithmica*, 52(1):3–18, 2008.

- [16] J. Kleinberg and E. Tardos. *Algorithm design*. Pearson Education India, 2006.
- [17] D. Kőnig. Graphok és matrixok. *Matematikai és Fizikai Lapok*, 38:116–119, 1931.
- [18] A. Mehta et al. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.
- [19] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized on-line matching. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 264–273. IEEE, 2005.
- [20] H. Nishimura and S. Kuroda. *A lost mathematician, Takeo Nakasawa: the forgotten father of matroid theory*. Springer Science & Business Media, 2009.
- [21] R. Rado. A theorem on independence relations. *The Quarterly Journal of Mathematics*, (1):83–89, 1942.
- [22] A. Srinivasan. Budgeted allocations in the full-information setting. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 247–253. Springer, 2008.
- [23] Y. Wang and S. C.-w. Wong. Matroid online bipartite matching and vertex cover. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 437–454. ACM, 2016.
- [24] H. Whitney. On the abstract properties of linear dependence. In *Hassler Whitney Collected Papers*, pages 147–171. Springer, 1992.