

Many-to-one Matchings with Lower Quotas: Algorithms and Complexity

Ashwin Arulselvan¹, Ágnes Cseh²(✉), Martin Groß², David F. Manlove³,
and Jannik Matuschke²

¹ Department of Management Science, University of Strathclyde,
Glasgow, Scotland, UK

ashwin.arulselvan@strath.ac.uk

² Institute for Mathematics, TU Berlin, Berlin, Germany

{cseh,gross,matuschke}@math.tu-berlin.de

³ School of Computing Science, University of Glasgow, Glasgow, Scotland, UK
David.Manlove@glasgow.ac.uk

Abstract. We study a natural generalization of the maximum weight many-to-one matching problem. We are given an undirected bipartite graph $G = (A \cup P, E)$ with weights on the edges in E , and with lower and upper quotas on the vertices in P . We seek a maximum weight many-to-one matching satisfying two sets of constraints: vertices in A are incident to at most one matching edge, while vertices in P are either unmatched or they are incident to a number of matching edges between their lower and upper quota. This problem, which we call *maximum weight many-to-one matching with lower and upper quotas* (WMLQ), has applications to the assignment of students to projects within university courses, where there are constraints on the minimum and maximum numbers of students that must be assigned to each project.

In this paper, we provide a comprehensive analysis of the complexity of WMLQ from the viewpoints of classic polynomial time algorithms, fixed-parameter tractability, as well as approximability. We draw the line between NP-hard and polynomially tractable instances in terms of degree and quota constraints and provide efficient algorithms to solve the tractable ones. We further show that the problem can be solved in polynomial time for instances with bounded treewidth; however, the corresponding runtime is exponential in the treewidth with the maximum upper quota u_{\max} as basis, and we prove that this dependence is necessary unless $\text{FPT} = \text{W}[1]$. Finally, we also present an approximation algorithm for the general case with performance guarantee $u_{\max} + 1$, which is asymptotically best possible unless $\text{P} = \text{NP}$.

Á. Cseh—Supported by the Deutsche Telekom Stiftung and by COST Action IC1205 on Computational Social Choice. Part of this work was carried out whilst visiting the University of Glasgow.

M. Groß—Supported by the DFG within project A07 of CRC TRR 154.

D.F. Manlove—Supported by EPSRC grant EP/K010042/1.

J. Matuschke—Supported by DAAD with funds of BMBF and the EU Marie Curie Actions.

Keywords: Maximum matching · Many-to-one matching · Project allocation · Inapproximability · Bounded treewidth

1 Introduction

Many university courses involve some element of team-based project work. A set of projects is available for a course and each student submits a subset of projects as acceptable. For each acceptable student–project pair (s, p) , there is a weight $w(s, p)$ denoting the *utility* of assigning s to p . The question of whether a given project can run is often contingent on the number of students assigned to it. Such quota constraints also arise in various other contexts involving the centralized formation of groups, including organizing activity groups at a leisure center, opening facilities to serve a community and coordinating rides within car-sharing systems. In these and similar applications, the goal is to maximize the utility of the assigned agents under the assumption that the number of participants for each open activity is within the activity’s prescribed limits.

We model this problem using a weighted bipartite graph $G = (A \dot{\cup} P, E)$, where the vertices in A represent *applicants*, while the vertices in P are *posts* they are applying to. So in the above student–project allocation example, A and P represent the students and projects respectively, and E represents the set of acceptable student–project pairs. The edge weights capture the cardinal utilities of an assigned applicant–post pair. Each post has a lower and an upper quota on the number of applicants to be assigned to it, while each applicant can be assigned to at most one post. In a feasible assignment, a post is either *open* or *closed*: the number of applicants assigned to an open post must lie between its lower and upper quota, whilst a closed post has no assigned applicant. The objective is to find a maximum weight many-to-one matching satisfying all lower and upper quotas. We denote this problem by WMLQ.

In this paper, we study the computational complexity of WMLQ from various perspectives: Firstly, in Sect. 2, we show that the problem can be solved efficiently if the degree of every post is at most 2, whereas the problem becomes hard as soon as posts with degree 3 are permitted, even when lower and upper quotas are all equal to the degree and every applicant has a degree of 2. Furthermore, we show the tractability of the case of pair projects, i.e., when all upper quotas are at most 2. Then, in Sect. 3, we study the fixed parameter tractability of WMLQ. To this end, we generalize the known dynamic program for maximum independent set with bounded treewidth to WMLQ. The running time of our algorithm is exponential in the treewidth of the graph, with u_{\max} , the maximum upper quota of any vertex, as the basis. This yields a fixed-parameter algorithm when parameterizing by both the treewidth and u_{\max} . We show that this exponential dependence on the treewidth cannot be completely separated from the remaining input by establishing a $W[1]$ -hardness result for WMLQ parameterized by treewidth. Finally, in Sect. 4, we discuss the approximability of the problem. We show that a simple greedy algorithm yields an approximation guarantee of $u_{\max} + 1$ for WMLQ and $\sqrt{|A|} + 1$ in the case of unit edge weights. We complement

these results by showing that these approximation factors are asymptotically best possible, unless $P = NP$.

Related work

Among various applications of centralized group formation, perhaps the assignment of medical students to hospitals has received the most attention. In this context, as well as others, the underlying model is a bipartite matching problem involving lower and upper quotas. The *Hospitals/Residents problem with Lower Quotas* (HRLQ) [4, 12] is a variant of WMLQ where applicants and posts have ordinal preferences over one another, and we seek a *stable matching* of residents to hospitals. Hamada et al. [12] considered a version of HRLQ where hospitals cannot be closed, whereas the model of Biró et al. [4] permitted hospital closures. Strategyproof mechanisms have also been studied in instances with ordinal preferences and no hospital closure [11].

The *Student/Project Allocation problem* [19, Sect. 5.6] models the assignment of students to projects offered by lecturers subject to upper and lower quota restrictions on projects and lecturers. Several previous papers have considered the case of ordinal preferences involving students and lecturers [1, 14, 20] but without allowing lower quotas. However two recent papers [15, 21] do permit lower quotas together with project closures, both in the absence of lecturer preferences. Monte and Tumennasan [21] considered the case where each student finds every project acceptable, and showed how to modify the classical Serial Dictatorship mechanism to find a Pareto optimal matching. Kamiyama [15] generalized this mechanism to the case where students need not find all projects acceptable, and where there may be additional restrictions on the sets of students that can be matched to certain projects. This paper also permits lower quotas and project closures, but our focus is on cardinal utilities rather than ordinal preferences.

The unit-weight version of WMLQ is closely related to the *D-matching problem* [8, 17, 26], a variant of graph factor problems [24]. In an instance of the *D-matching problem*, we are given a graph G , and a domain of integers is assigned to each vertex. The goal is to find a subgraph G' of G such that every vertex has a degree in G' that is contained in its domain. Lovász [16] showed that the problem of deciding whether such a subgraph exists is NP-complete, even if each domain is either $\{1\}$ or $\{0, 3\}$. On the other hand, some cases are tractable. For example, if for each domain D , the complement of D contains no consecutive integers, the problem is polynomially solvable [26]. As observed in [25], *D-matchings* are closely related to *extended global cardinality constraints* and the authors provide an analysis of the fixed-parameter tractability of a special case of the *D-matching problem*; see Sect. 3 for details.

The problem that we study in this paper corresponds to an optimization version of the *D-matching problem*. We consider the special case where G is bipartite and the domain of each applicant vertex is $\{0, 1\}$, whilst the domain of each post vertex p is $\{0\} \cup \{\ell(p), \dots, u(p)\}$, where $\ell(p)$ and $u(p)$ denote the lower and upper quotas of p respectively. Since the empty matching is always

feasible in our case, our aim is to find a domain-compatible subgraph G' such that the total weight of the edges in G' is maximum.

2 Degree- and Quota-restricted Cases

First, we provide a formal definition of the maximum weight many-to-one matching problem with lower quotas (WMLQ). Then, we characterize the complexity of the problem in terms of degree constraints on the two vertex sets: applicants and posts. At the end, we discuss the case of bounded upper quota constraints.

2.1 Problem Definition

In our problem, a set of applicants A and a set of posts P are given. A and P constitute the two vertex sets of an undirected bipartite graph $G = (V, E)$ with $V = A \dot{\cup} P$. For a vertex $v \in V$ we denote by $\delta(v) = \{\{v, w\} \in E : w \in V\}$ the set of edges incident to v and by $\Gamma(v) = \{w \in V : \{v, w\} \in E\}$ the *neighborhood* of v , i.e., the set of vertices that are adjacent to v . For a subset of vertices $V' \subset V$, we define $\delta(V') = \bigcup_{v \in V'} \delta(v)$. Each edge carries a *weight* $w : E \rightarrow \mathbb{R}_{\geq 0}$, representing the utility of the corresponding assignment. Each post is equipped with a *lower quota* $\ell : P \rightarrow \mathbb{Z}_{\geq 0}$ and an *upper quota* $u : P \rightarrow \mathbb{Z}_{\geq 0}$ so that $\ell(p) \leq u(p)$ for every $p \in P$. These functions bound the number of admissible applicants for the post (independent of the weight of the corresponding edges). Furthermore, every applicant can be assigned to at most one post. Thus, an *assignment* is a subset $M \subseteq E$ of the edges such that $|\delta(a) \cap M| \leq 1$ for every applicant $a \in A$ and $|\delta(p) \cap M| \in \{0, \ell(p), \ell(p) + 1, \dots, u(p)\}$ for every $p \in P$. A post is said to be *open* if the number of applicants assigned to it is greater than 0, and *closed* otherwise. The *size* of an assignment M , denoted $|M|$, is the number of assigned applicants, while the *weight* of M , denoted $w(M)$, is the total weight of the edges in M , i.e., $w(M) = \sum_{e \in M} w(e)$. The goal is to find an assignment of maximum weight.

Remark 1. Note that when not allowing closed posts, the problem immediately becomes tractable. It is easy to see this in the unweighted case as any algorithm for maximum flow with lower capacities can be used to determine an optimal solution in polynomial time. This problem can be easily reduced to the classical maximum flow problem. The method can be naturally extended to the weighted case as the flow based linear program has integral extreme points due to its total unimodularity property.

Problem 1. WMLQ

Input: $\mathcal{I} = (G, w, \ell, u)$; a bipartite graph $G = (A \dot{\cup} P, E)$ with edge weights w .

Task: Find an assignment of maximum weight.

If $w = 1$ for all $e \in E$, we refer to the problem as MLQ.

Some trivial simplification of the instance can be executed right at start. If $u(p) > |\Gamma(p)|$ for a post p , then $u(p)$ can be replaced by $|\Gamma(p)|$. On the other

hand, if $\ell(p) > |\Gamma(p)|$, then post p can immediately be deleted, since no feasible solution can satisfy the lower quota condition. Moreover, posts with $\ell(p) = 1$ behave identically to posts without a lower quota. From now on we assume that the instances have already been simplified this way.

2.2 Degree-Restricted Cases

In this subsection, we will consider $\text{WMLQ}(i, j)$, a special case of WMLQ , in which we restrict us to instances in which every applicant submits at most i applications and every post receives at most j applications. In order to establish our first result, we reduce the maximum independent set problem (MIS) to MLQ . In MIS, a graph with n vertices and m edges is given and the task is to find an independent vertex set of maximum size. MIS is not approximable within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$, unless $\text{P} = \text{NP}$ [29]. The problem remains APX-complete even for cubic (3-regular) graphs [2].

Theorem 1. $\text{MLQ}(2,3)$ is APX-complete.

Proof. First of all, $\text{MLQ}(2,3)$ is in APX because feasible solutions are of polynomial size and the problem has a 4-approximation (see Theorem 7).

To each instance \mathcal{I} of MIS on cubic graphs we create an instance \mathcal{I}' of MLQ such that there is an independent vertex set of size at least K in \mathcal{I} if and only if \mathcal{I}' admits an assignment of size at least $3K$, yielding an approximation-preserving reduction. The construction is as follows. To each of the n vertices of graph G in \mathcal{I} , a post with upper and lower quota of 3 is created. The m edges of G are represented as m applicants in \mathcal{I}' . For each applicant $a \in A$, $|\Gamma(a)| = 2$ and $\Gamma(a)$ comprises the two posts representing the two end vertices of the corresponding edge. Since we work on cubic graphs, $|\Gamma(p)| = 3$ for every post $p \in P$.

First we show that an independent vertex set of size K can be transformed into an assignment of at least $3K$ applicants. All we need to do is to open a post with its entire neighborhood assigned to it if and only if the vertex representing that post is in the independent set. Since no two posts stand for adjacent vertices in G , their neighborhoods do not intersect. Moreover, the assignment assigns exactly three applicants to each of the K open posts.

To establish the opposite direction, let us assume that an assignment of cardinality at least $3K$ is given. The posts' upper and lower quota are both set to 3, therefore, the assignment involves at least K open posts. No two of them can represent adjacent vertices in G , because then the applicant standing for the edge connecting them would be assigned to both posts at the same time.

The reduction given here is an L -reduction [23] with constants $\alpha = \beta = 3$. Since $\text{MLQ}(2,3)$ belongs to APX and MIS is APX-complete in cubic graphs, it follows that $\text{MLQ}(2,3)$ is APX-complete. \square

So far we have established that if $|\Gamma(a)| \leq 2$ for every applicant $a \in A$ and $|\Gamma(p)| \leq 3$ for every post $p \in P$, then MLQ is NP-hard. In the following, we also show that these restrictions are the tightest possible. If $|\Gamma(p)| \leq 2$ for every post $p \in P$, then a maximum weight matching can be found efficiently, regardless of $|\Gamma(a)|$. Note that the case $\text{WMLQ}(1, \infty)$ is trivially solvable.

Theorem 2. $\text{WMLQ}(\infty, 2)$ is solvable in $O(n^2 \log n)$ time, where $n = |A| + |P|$.

Proof. After executing the simplification steps described after the problem definition, we apply two more changes to derive our helper graph H . Firstly, if $\ell(p) = 0$, $u(p) = 2$ and $|\Gamma(p)| = 2$, we separate p 's two edges, splitting p into two posts with upper quota 1. After this step, all posts with $u(p) = 2$ also have $\ell(p) = 2$. All remaining vertices are of upper quota 1. Then, we substitute all edge pairs of posts with $\ell(p) = u(p) = 2$ with a single edge connecting the two applicants. This edge will carry the weight equal to the sum of the weights of the two deleted edges.

Clearly, any matching in H translates into an assignment of the same weight in G and vice versa. Finding a maximum weight matching in a general graph with n vertices and m edges can be done in $O(n(m + n \log n))$ time [10], which reduces to $O(n^2 \log n)$ in our case.

2.3 Quota-Restricted Cases

In this section, we address the problem of WMLQ with bounded upper quotas. Note that Theorem 1 already tells us that the case of $u(p) \leq 3$ for all posts $p \in P$ is NP-hard to solve. We will now settle the complexity of the only remaining case, where we have instances with every post $p \in P$ having an arbitrary degree and $u(p) \leq 2$. This setting models posts that need to be assigned to pairs of applicants.

The problem is connected to various known problems in graph theory, one of them being the *S-path packing problem*. In that problem, we are given a graph with a set of terminal vertices S . The task is to pack the highest number of vertex-disjoint paths so that each path starts and ends at a terminal vertex, while all its inner vertices are non-terminal. The problem can be solved in $O(n^{2.38})$ time [7, 27] with the help of matroid matching [18]. An instance of MLQ with $\ell(p) = u(p) = 2$ for every post $p \in P$ corresponds to an S -path packing instance with $S = A$. The highest number of vertex-disjoint paths starting and ending in A equals half of the cardinality of a maximum assignment. Thus, MLQ with $\ell(p) = u(p) = 2$ can also be solved in $O(n^{2.38})$ time. On the other hand, there is no straightforward way to model posts with $u(p) = 1$ in S -path packing and introducing weights to the instances also seems to be a challenging task. Some progress has been made for weighted edge-disjoint paths, but to the best of our knowledge the question is unsettled for vertex-disjoint paths [13].

In the full version of the paper [3] we present a solution for the general case WMLQ with $u(p) \leq 2$. Our algorithm is based on f -factors of graphs [9].

Theorem 3. WMLQ with $u(p) \leq 2$ for every $p \in P$ can be solved in $O(nm + n^2 \log n)$ time, where $n = |V|$ and $m = |E|$.

3 Bounded treewidth graphs

In this section, we investigate WMLQ from the point of view of fixed-parameter tractability and analyze how efficiently the problem can be solved for instances with a bounded treewidth.

Fixed-parameter tractability. This field of complexity theory is motivated by the fact that in many applications of optimization problems certain input parameters stay small even for large instances. A problem, parameterized by a parameter k , is fixed-parameter tractable (FPT) if there is an algorithm solving it in time $f(k) \cdot \phi(n)$, where $f : \mathbb{R} \rightarrow \mathbb{R}$ is a function, ϕ is a polynomial function, and n is the input size of the instance. Note that this definition not only requires that the problem can be solved in polynomial time for instances where k is bounded by a constant, but also that the dependence of the running time on k is separable from the part depending on the input size. On the other hand, if a problem is shown to be $W[1]$ – *hard*, then the latter property can only be fulfilled if $FPT = W[1]$, which would imply $NP \subseteq DTIME(2^{o(n)})$. For more details on fixed-parameter algorithms see, e.g., [22].

Treewidth. In case of WMLQ we focus on the parameter *treewidth*, which, on an intuitive level, describes the likeness of a graph to a tree. A *tree decomposition* of graph G consists of a tree whose nodes—also called *bags*—are subsets of $V(G)$. These must satisfy the following three requirements.

1. Every vertex of G belongs to at least one bag of the tree.
2. For every edge $\{a, p\} \in E(G)$, there is a bag containing both a and p .
3. If a vertex in $V(G)$ occurs in two bags of the tree, then it also occurs in all bags on the unique path connecting them.

The *width* of a tree decomposition with a set of bags B is $\max_{b \in B} |b| - 1$. The *treewidth* of a graph G , $\text{tw}(G)$, is the smallest width among all tree decompositions of G . It is well known that a tree decomposition of smallest width can be found by a fixed-parameter algorithm when parameterized by $\text{tw}(G)$ [5].

In the following, we show that WMLQ is fixed-parameter tractable when parameterized simultaneously by the treewidth and u_{\max} , whereas it remains $W[1]$ -hard when only parameterized by the treewidth. A similar study of the fixed-parameter tractability of the related *extended global cardinality constraint problem* (EGCC) was conducted in [25]. EGCC corresponds to the special case of the D -matching problem where the graph is bipartite and on one side of the bipartition all vertices have the domain $\{1\}$. Differently from WMLQ, EGCC is a feasibility problem (note that the feasibility version of WMLQ is trivial, as the empty assignment is always feasible). The authors of [25] provide a fixed-parameter algorithm for EGCC when parameterized simultaneously by the treewidth of the graph and the maximum domain size, and they show that the problem is $W[1]$ -hard when only parameterized by the treewidth. These results mirror our results for WMLQ, and indeed both our FPT-algorithm for WMLQ and the one in [25] are extensions of the same classic dynamic program for the underlying maximum independent set problem. However, our hardness result uses a completely different reduction than the one in [25]. The latter makes heavy use of the fact that the domains can be arbitrary sets, whereas in WMLQ, we are confined to intervals.

Theorem 4. WMLQ can be solved in time $O(T + (u_{\max})^{3 \text{tw}(G)} |E|)$, where T is the time needed for computing a tree decomposition of G . In particular, WMLQ can be solved in polynomial time when restricted to instances of bounded treewidth, and WMLQ parameterized by $\max\{\text{tw}(G), u_{\max}\}$ is fixed-parameter tractable.

The algorithmic proof of Theorem 4 can be found in the full version of the paper [3]. While our algorithm runs in polynomial time for bounded treewidth, the degree of the polynomial depends on the treewidth the algorithm only becomes a fixed-parameter algorithm when parameterizing by treewidth and u_{\max} simultaneously. We will now show by a reduction from MINIMUM MAXIMUM OUTDEGREE that this dependence is necessary under the assumption that $\text{FPT} \neq \text{W}[1]$.

Problem 2. MINIMUM MAXIMUM OUTDEGREE

Input: A graph $G = (V, E)$, edge weights $w : E \rightarrow \mathbb{Z}_+$ encoded in unary, a degree-bound $r \in \mathbb{Z}_+$.

Task: Find an orientation D of G such that $\sum_{e \in \delta_D^+(v)} w(e) \leq r$ for all $v \in V$, where $\delta_D^+(v)$ stands for the set of edges oriented so that their tail is v .

Theorem 5 (Theorem 5 from [28]). MINIMUM MAXIMUM OUTDEGREE is $\text{W}[1]$ -hard when parameterized by treewidth.

Theorem 6. MLQ is $\text{W}[1]$ -hard when parameterized by treewidth, even when restricted to instances where $\ell(p) \in \{0, u(p)\}$ for every $p \in P$.

Proof. Given an instance $(G = (V, E), w, r)$ of MINIMUM MAXIMUM OUTDEGREE, we construct an instance $(G' = (A \dot{\cup} P, E'), \ell, u)$ of MLQ as follows. For every vertex $v \in V$ we introduce a post $p_v \in P$ and let $\ell(p_v) = 0$ and $u(p_v) = r$. Furthermore, for every edge $e = \{v, v'\} \in E$, we introduce two posts $p_{e,v}$ and $p_{e,v'}$ with $\ell(p_{e,v}) = \ell(p_{e,v'}) = u(p_{e,v}) = u(p_{e,v'}) = w(e) + 1$, and $2w(e) + 1$ applicants $a_{e,v}^1, \dots, a_{e,v}^{w(e)}, a_{e,v'}^1, \dots, a_{e,v'}^{w(e)}, z_e$, for which we introduce the edges $\{p_v, a_{e,v}^i\}$, $\{a_{e,v}^i, p_{e,v}\}$, $\{p_{v'}, a_{e,v'}^i\}$, and $\{a_{e,v'}^i, p_{e,v'}\}$ for $i \in \{1, \dots, w(e)\}$ as well as $\{p_{e,v}, z_e\}$ and $\{z_e, p_{e,v'}\}$.

We show that the constructed instance has a solution serving all applicants if and only if the MINIMUM MAXIMUM OUTDEGREE instance has an orientation respecting the bound on the outdegree.

First assume there is an orientation D of G with maximum outdegree at most r . Then consider the assignment that assigns for every oriented edge $(v, v') \in D$ the $w(e)$ applicants $a_{e,v}^i$ to p_v and the $w(e) + 1$ applicants $a_{e,v'}^i$ and z_e to $p_{e,v'}$. As the weighted outdegree of vertex v is at most r , every post p_v gets assigned at most $r = u(p_v)$ applicants.

Now assume M is a feasible assignment of applicants to posts serving every applicant. In particular, for every edge $e = \{v, v'\} \in E$, applicant z_e is assigned to either $p_{e,v}$ or $p_{e,v'}$ and exactly one of these two posts is open because the lower bound of $w(e) + 1$ can only be met if z_e is assigned to the respective post. If $p_{e,v}$ is open then all $w(e)$ applicants $a_{e,v'}^i$ are assigned to $p_{v'}$ and none of the applicants $a_{e,v}^i$ is assigned to p_v , and vice versa if $p_{e,v'}$ is open. Consider the

orientation obtained by orienting every edge e from v to v' if and only if $p_{e,v}$ is open. By the above observations, the weighted outdegree of vertex v corresponds to the number of applicants assigned to post p_v , which is at most r .

Finally, note that G' can be constructed in time polynomial in the input size of the MINIMUM MAXIMUM OUTDEGREE instance as the weights are encoded in unary there. Furthermore, the treewidth of G' is at most $\max\{\text{tw}(G), 3\}$. To see this, start with a tree decomposition of G and identify each vertex $v \in V$ with the corresponding post p_v . For every edge $e = \{v, v'\} \in E$, there is a bag B with $p_v, p'_v \in B$. We add the new bag $B_e = \{p_v, p'_v, p_{e,v}, p_{e,v'}\}$ as a child to B . We further add the bags $B_{z_e} = \{p_{e,v}, p_{e,v'}, z_e\}$, $B_{a_{e,v}^i} = \{p_v, p_{e,v}, a_{e,v}^i\}$ and $B_{a_{e,v'}^i} = \{p_{v'}, p_{e,v'}, a_{e,v'}^i\}$ for $i \in \{1, \dots, w(e)\}$ as children to B_e . Observe that the tree of bags generated by this construction is a tree decomposition. Furthermore, since we did not increase the size of any of the existing bags and added only bags of size at most 4 the treewidth of G' is at most $\max\{\text{tw}(G), 3\}$. \square

4 Approximation

Having established the hardness of WMLQ even for very restricted instances in Theorem 1, we turn our attention towards approximability. In this section, we give an approximation algorithm and corresponding inapproximability bounds expressed in terms of $|A|, |P|$ and upper quotas in the graph.

The method, which is formally listed in Algorithm 1, is a simple greedy algorithm. We say a post p is *admissible* if it is not yet open and $|\Gamma(p)| \geq \ell(p)$. The algorithm iteratively opens an admissible post maximizing the assignable weight, i.e., it finds a post $p' \in P$ and a set A' of applicants in its neighborhood $\Gamma(p')$ with $\ell(p') \leq |A'| \leq u(p')$ such that $\sum_{a \in A'} w(a, p')$ is maximized among all such pairs. It then removes the assigned applicants from the graph (potentially rendering some posts inadmissible) and re-iterates until no admissible post is left.

Algorithm 1. Greedy algorithm for WMLQ

Initialize $P_0 = \{p \in P : |\Gamma(p)| \geq \ell(p)\}$.

Initialize $A_0 = A$.

while $P_0 \neq \emptyset$ **do**

Find a pair $p' \in P_0$ and $A' \subseteq \Gamma(p')$ with $|A'| \leq u(p')$ such that $\sum_{a \in A'} w(a, p')$ is maximized among all such pairs.

Open p' and assign all applicants in A' to it.

Remove p' from P_0 and remove the elements of A' from A_0 .

for $p \in P_0$ with $\ell(p) > |\Gamma(p) \cap A_0|$ **do**

Remove p from P_0 .

end for

end while

In the full version of the paper [3] we give a tight analysis of the algorithm, establishing approximation guarantees in terms of the number of posts $|P|$, number of applicants $|A|$, and the maximum upper quota $u_{\max} := \max_{p \in P} u(p)$ over

all posts. We also provide two examples that show that our analysis of the greedy algorithm is tight for each of the described approximation factors. We further show there that the approximation ratios given above for WMLQ are almost tight from the point of view of complexity theory.

We point out a reduction from WMLQ to the set packing problem here. The elements in the universe of the set packing problem would be $A \cup P$. For each post p and for each subset $S \subset \Gamma(p)$, such that $l(p) \leq |S| \leq u(p)$, we create a set $S \cup \{p\}$ for the set packing instance. However, if the difference between upper and lower quota is not bounded, this would create an exponential sized input for the set packing problem and we could only employ an oracle based algorithm known for set packing problem to solve WMLQ. The greedy algorithm known for set packing problem [6] can be made to work in a fashion similar to the algorithm presented above.

Theorem 7. *Algorithm 1 is an α -approximation algorithm for WMLQ with $\alpha = \min\{|P|, |A|, u_{\max} + 1\}$. Furthermore, for MLQ, Algorithm 1 is a $\sqrt{|A|} + 1$ -approximation algorithm. It can be implemented to run in time $O(|E| \log |E|)$.*

Theorem 8. *MLQ is not approximable within a factor of $|P|^{1-\varepsilon}$ or $\sqrt{|A|}^{1-\varepsilon}$ or $u_{\max}^{1-\varepsilon}$ for any $\varepsilon > 0$, unless $P = NP$, even when restricting to instances where $l(p) = u(p)$ for every $p \in P$ and $|\Gamma(a)| \leq 2$ for every $a \in A$.*

Acknowledgements. We would like to thank András Frank and Kristóf Bérczi for their observations that led us to Theorem 3 and the anonymous reviewers for their valuable comments, which have helped to improve the presentation of this paper.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, duplication, adaptation, distribution, and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license, and any changes made are indicated. The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt, or reproduce the material.

References

1. Abraham, D.J., Irving, R.W., Manlove, D.F.: Two algorithms for the student-project allocation problem. *J. Discrete Algorithms* **5**(1), 79–91 (2007)
2. Alimonti, P., Kann, V.: Some APX-completeness results for cubic graphs. *Theoret. Comput. Sci.* **237**(1–2), 123–134 (2000)
3. Arulselvan, A., Cseh, Á., Groß, M., Manlove, D.F., Matuschke, J.: Many-to-one matchings with lower quotas: algorithms and complexity (2015). CoRR, [abs/1412.0325](https://arxiv.org/abs/1412.0325)

4. Biró, P., Fleiner, T., Irving, R.W., Manlove, D.F.: The College Admissions problem with lower and common quotas. *Theoret. Comput. Sci.* **411**, 3136–3153 (2010)
5. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25**(6), 1305–1317 (1996)
6. Chandra, B., Halldórsson, M.M.: Greedy local improvement and weighted set packing approximation. *J. Algorithms* **39**(2), 223–240 (2001)
7. Cheung, H.Y., Lau, L.C., Leung, K.M.: Algebraic algorithms for linear matroid parity problems. *ACM Trans. Algorithms (TALG)* **10**(3), 10:1–10:26 (2014)
8. Cornuéjols, G.: General factors of graphs. *J. Comb. Theor. Ser. B* **45**(2), 185–198 (1988)
9. Gabow, H.: An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In: *Proceedings of STOC 1983: the 15th Annual ACM Symposium on Theory of Computing*, pages 448–456. ACM (1983)
10. Gabow, H.: Data structures for weighted matching and nearest common ancestors with linking. In: *Proceedings of SODA 1990: the 1st ACM-SIAM Symposium on Discrete Algorithms*, pp. 434–443. ACM-SIAM (1990)
11. Goto, M., Hashimoto, N., Iwasaki, A., Kawasaki, Y., Ueda, S., Yasuda, Y., Yokoo, M.: Strategy-proof matching with regional minimum quotas. In: *Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2014*, pp. 1225–1232, Richland (2014)
12. Hamada, K., Iwama, K., Miyazaki, S.: The Hospitals/Residents problem with lower quotas. *Algorithmica*, pp. 1–26 (2014)
13. Hirai, H., Pap, G.: Tree metrics and edge-disjoint S -paths. *Math. Program.* **147**(1–2), 81–123 (2014)
14. Iwama, K., Miyazaki, S., Yanagisawa, H.: Improved approximation bounds for the student-project allocation problem with preferences over projects. *J. Discrete Algorithms* **13**, 59–66 (2012)
15. Kamiyama, N.: A note on the serial dictatorship with project closures. *Oper. Res. Lett.* **41**, 559–561 (2013)
16. Lovász, L.: On the structure of factorizable graphs. *Acta Mathematica Academiae Scientiarum Hungaricae* **23**(1–2), 179–195 (1972)
17. Lovász, L.: Antifactors of graphs. *Periodica Mathematica Hungarica* **4**(2–3), 121–123 (1973)
18. Lovász, L.: Matroid matching and some applications. *J. Comb. Theor. Ser. B* **28**(2), 208–236 (1980)
19. Manlove, D.F.: *Algorithmics of Matching Under Preferences*. World Scientific, Singapore (2013)
20. Manlove, D.F., O'Malley, G.: Student project allocation with preferences over projects. *J. Discrete Algorithms* **6**, 553–560 (2008)
21. Monte, D., Tumennasan, N.: Matching with quorums. *Econ. Lett.* **120**, 14–17 (2013)
22. Niedermeier, R.: *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, Oxford (2006)
23. Papadimitriou, C., Yannakakis, M.: Optimization, approximation and complexity classes. *J. Comput. Syst. Sci.* **43**(3), 425–440 (1991)
24. Plummer, M.: Graph factors and factorization: 1985–2003: a survey. *Discrete Math.* **307**(7–8), 791–821 (2007)
25. Samer, M., Szeider, S.: Tractable cases of the extended global cardinality constraint. *Constraints* **16**, 1–24 (2011)
26. Sebő, A.: General antifactors of graphs. *J. Comb. Theor. Ser. B* **58**(2), 174–184 (1993)

27. Sebő, A., Szegő, L.: The path-packing structure of graphs. In: Bienstock, D., Nemhauser, G.L. (eds.) IPCO 2004. LNCS, vol. 3064, pp. 256–270. Springer, Heidelberg (2004)
28. Szeider, S.: Not so easy problems for tree decomposable graphs (2011). CoRR, [abs/1107.1177](https://arxiv.org/abs/1107.1177)
29. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. *Theor. Comput.* **3**(6), 103–128 (2007)