MAGYAR TUDOMÁNYOS AKADÉMIA
Közgazdaság- és Regionális Tudományi Kutatóközpont

MTA KRTK KTI

Centre for Economic and Regional Studies
HUNGARIAN ACADEMY OF SCIENCES

MŰHELYTANULMÁNYOK DISCUSSION PAPERS

MT-DP – 2017/24

Matchings with lower quotas: Algorithms and complexity

ASHWIN ARULSELVAN – ÁGNES CSEH – MARTIN GROß

DAVID F. MANLOVE – JANNIK MATUSCHKE

INSTITUTE OF ECONOMICS, CENTRE FOR ECONOMIC AND REGIONAL STUDIES,
HUNGARIAN ACADEMY OF SCIENCES - BUDAPEST, 2017

Discussion papers
MT-DP – 2017/24

Institute of Economics, Centre for Economic and Regional Studies,
Hungarian Academy of Sciences

Matchings with lower quotas: Algorithms and complexity

Authors:

Ashwin Arulselvan
Department of Management Science, Sir William Duncan Building, University of Strathclyde
E-mail: ashwin.arulselvan@strath.ac.uk

Ágnes Cseh,
research fellow
Institute of Economics, Research Centre for Economic and Regional Studies
Hungarian Academy of Sciences,
and Corvinus University of Budapest
E-mail: cseh.agnes@krtk.mta.hu

Martin Groß
Institute for Mathematics, Technische Universität Berlin
E-mail: gross@math.tu-berlin.de

David F. Manlove
School of Computing Science, Sir Alwyn Williams Building, University of Glasgow
E-mail: David.Manlove@glasgow.ac.uk

Jannik Matuschke
TUM School of Management, Technische Universiät München
E-mail: jannik.matuschke@tum.de

September 2017

ISBN 978-615-5754-14-2
ISSN 1785 377X

Alsó kvótás párosítások: algoritmusok és komplexitás

Ashwin Arulselvan - Ágnes Cseh - Martin Groß - David F. Manlove - Jannik Matuschke

Összefoglaló

A maximum súlyú párosítás egy természetes általánosítását tanulmányozzuk cikkünkben. Adott egy irányítatlan páros gráf G = (A u P, E), amelyben minden él súlyozott és minden P-beli csúcson egy alsó és felső kvóta van. Célunk egy maximális súlyú b-párosítás keresése, amiben az A-beli csúcsok foka legfeljebb 1, míg a P-belieké vagy 0 vagy a csúcs alsó és felső kvótája közt van. Ezt a problémát WMLQ-nak nevezzük. Fő motivációja egyetemi csoportbeosztásokban keresendő, amelyeknél csoportonként adott egy minimum- és maximumlétszám. Cikkünkben lefolytatjuk a WMLQ-probléma bonyolultságának teljes vizsgálatát, kitérve mind a klasszikus polinomiális algoritmusokra, mind a fix-paraméter kezelhetőségre, mind a közelítő algoritmusokra. Meghúzzuk a NP-nehéz polinomiális esetek közti választóvonalat maximális fokszámban és kvótában kifejezve és hatékony algoritmusokat adunk a megoldható esetekre. Ezen felül alacsony favastagságú gráfokra is bemutatunk egy polinomiális algoritmust. Egy közelítő algoritmust is adunk WMLQ-ra, amelyről azt is belátjuk, hogy a lehető legjobb. Végül kifejtjük, hogy eredményeink hogyan általánosíthatók sok-a-sokhoz párosításokra.

Kulcsszavak: maximum párosítás, sok-az-egyben párosítás, projektallokáció, közelíthetőség, korlátos favastagság

JEL-kód: C63, C78

Matchings with lower quotas: Algorithms and complexity

Ashwin Arulselvan · Ágnes Cseh · Martin Groß · David F. Manlove · Jannik Matuschke

Abstract We study a natural generalization of the maximum weight many-to-one matching problem. We are given an undirected bipartite graph $G = (A \dot\cup P, E)$ 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

A preliminary version of this paper appeared at the 26th International Symposium on Algorithms and Computation (ISAAC 2015). The authors were supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2016-3), COST Action IC1205 on Computational Social Choice, DFG within project A07 of CRC TRR 154, EPSRC grant EP/K010042/1, DAAD with funds of BMBF and the EU Marie Curie Actions. Part of this work was carried out whilst Á. Cseh was visiting the University of Glasgow.

A. Arulselvan
Department of Management Science, Sir William Duncan Building, University of Strathclyde, Glasgow, G4 0GE, UK
E-mail: ashwin.arulselvan@strath.ac.uk

Á. Cseh
Institute of Economics, Hungarian Academy of Sciences, Budaörsi út 45., 1112 Budapest, Hungary
Tel.: +36 1309 2652
Fax: +36 1309 2650
E-mail: cseh.agnes@krtk.mta.hu

M. Groß
Institute for Mathematics, TU Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany
E-mail: gross@math.tu-berlin.de

D.F. Manlove
School of Computing Science, Sir Alwyn Williams Building, University of Glasgow, Glasgow G12 8QQ, UK
E-mail: David.Manlove@glasgow.ac.uk

J. Matuschke
TUM School of Management, Technische Universität München, Arcisstraße 21, 80333 Munich, Germany
E-mail: jannik.matuschke@tum.de

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 classical 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 FPT = W[1]. The approximability of WMLQ is also discussed: we present an approximation algorithm for the general case with performance guarantee $u_{\max} + 1$, which is asymptotically best possible unless P = NP. Finally, we elaborate on how most of our positive results carry over to matchings in arbitrary graphs with lower quotas.

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 team-based activities 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. We begin by defining the problem formally in Section 2. Then in Section 3, we show that WMLQ 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. In Section 4, 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 parameterized 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 Section 5, 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 $\mathsf{P} = \mathsf{NP}$. We briefly comment on the generalizability our aforementioned results in Section 6 for matchings in arbitrary graphs with lower quotas.

1.1 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, 14] 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. [14] 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 closures [9, 12, 13].

The *Student / Project Allocation problem* [24, Section 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, 16, 25] but without allowing lower quotas. However two recent papers [18, 26] do permit lower quotas together with project closures, both in the absence of lecturer preferences. Monte and Tumennasan [26] 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 [18] 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. Cardinal utilities can facilitate a more flexible representation of preferences and they occur naturally in various matching problems, such as the solution of symmetric indefinite systems [31], the organization of Chess tournaments [28] or car sharing [15]. Moreover, various methods for converting ordinal preferences into cardinal utilities are known, such as the normalized Borda score [30].

The unit-weight version of WMLQ is also closely related to the *D-matching problem* [8, 23, 33], a variant of graph factor problems [29]. 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 [22] 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 [33]. As observed in [32], D-matchings are closely related to *extended global cardinality constraints* and the authors provided an analysis of the fixed-parameter tractability of a special case of the D-matching problem; see Section 4 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 Problem definition

In this section we provide a formal definition of the maximum weight many-to-one matching problem with lower quotas (WMLQ).

Basic notation Let $G = (V, E)$ be a graph. For a subset of vertices $U \subseteq V$ we denote by $\delta(U) = \{\{v, w\} \in E : v \in U, w \in V \setminus U\}$ the set of edges incident to exactly one vertex in U. For a vertex $v \in V$, we write $\delta(v) = \delta(\{v\})$, and for a subset of edges $F \subseteq E$ we write $\deg_F(v) = |\delta(v) \cap F|$. By $\Gamma(v) = \{w \in V : \{v, w\} \in E\}$ we denote the *neighborhood* of v, i.e., the set of vertices that are adjacent to v.

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 $V = A \dot\cup P$ and E represents the set of acceptable applicant-post pairs. Each edge carries a *weight* $w : E \to \mathbb{R}_{\geq 0}$, representing the utility of the corresponding assignment. The set of posts is equipped with functions $\ell : P \to \mathbb{Z}_{\geq 0}$ and $u : P \to \mathbb{Z}_{\geq 0}$ such that $\ell(p) \leq u(p)$ for every $p \in P$. Here $\ell(p)$ is called the *lower quota* of p and $u(p)$ is called the *upper quota* of 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 $|\delta(p) \cap M| \in \{0, \ell(p), \ell(p)+1, ..., u(p)\}$ for every $p \in P$. With respect to an assignment M, 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. Formally, we define the WMLQ problem as follows.

Problem 1 WMLQ
Input: $\mathcal{I} = (G, w, \ell, u)$*; a bipartite graph* $G = (A \dot\cup P, E)$ *with edge weights* w*, lower quotas* ℓ *and upper quotas* u*.*
Task: Find an assignment of maximum weight.
If $w(e) = 1$ *for all* $e \in E$*, we refer to the problem as* MLQ*.*

Remark 1 Note that when lower quotas are present but posts cannot be closed, we obtain the *Degree Constrained Subgraph Problem* (DCS); a maximum weight matching in an instance of DCS can be found in polynomial time [10]. In the special case that lower quotas are absent, DCS is referred to as the *Upper Degree Constrained Subgraph Problem* (UDCS) [10].

Some trivial simplification of the instance can be executed right at the 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, a post p with $\ell(p) = 1$ behaves identically to the case that $\ell(p) = 0$, so we assume that no post p has $\ell(p) = 1$. From now on we assume that the instances have already been simplified this way.

3 Degree- and quota-restricted cases

In this section we characterize the complexity of WMLQ in the presence of upper bounds placed on vertex degrees or the posts' upper quotas. Section 3.1 deals with degree-restricted cases, whilst Section 3.2 studies cases involving bounded upper quotas.

3.1 Degree-restricted cases

In this subsection we will consider WMLQ(i, j), the special case of WMLQ in which $|\Gamma(a)| \leq i$ for all $a \in A$, and $|\Gamma(p)| \leq j$ for all $p \in P$. That is, 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 P = NP [35]. The problem remains APX-complete even for cubic (3-regular) graphs [2].

Theorem 1 MLQ(2,3) *is* APX*-complete.*

Proof First of all, MLQ(2,3) is in APX because the problem has a 4-approximation that can be found in polynomial time (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.

Note that every solution of the constructed instance of MLQ serves an integer multiple of 3 applicants. In particular, the MLQ instance has a solution serving $3K$ applicants if and only if there is an independent set of size K in the MIS instance. Hence, this reduction preserves the approximation factors. Since MLQ(2,3) belongs to APX and MIS is APX-complete in cubic graphs, it follows that MLQ(2,3) is APX-complete. □

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 WMLQ$(1,\infty)$ is trivially solvable.

Theorem 2 WMLQ(∞,2) *is solvable in* $O(n^2 \log n)$ *time, where* $n = |A|+|P|$.

Proof After executing the simplification steps described at the end of Section 2, 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 $G = (V, E)$ can be done in $O(|V|(|E| + |V| \log |V|))$ time [11], which reduces to $O(|V|^2 \log |V|)$ in our case. □

3.2 Quota-restricted cases

In this section, we consider restrictions 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 none, one or pairs of applicants.

Here we present a solution for WMLQ with $u(p) \leq 2$. Our algorithm is based on f-factors of graphs. In the *f-factor problem*, a graph G and a function $f : V \to \mathbb{Z}_{\geq 0}$ is given. A set of edges $F \subseteq E$ is called an *f-factor* if $\deg_F(v) = f(v)$ for every $v \in V$, where $\deg_F(v)$, as per our earlier definition, is the degree of v in the graph (V, F). Constructing an f-factor of maximum weight in a graph with n vertices and m edges or proving that none exists can be done in $O(\phi(m+n \log n))$ time, where ϕ is the sum of all f-values in the graph [10, 11].

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* $m = |E|$.

Proof We partition P into P_1 and $P \setminus P_1$, where P_1 denotes the set of posts with $u(p) = 1$. For posts in $P \setminus P_1$ we can assume that $\ell(p) = u(p) = 2$ for every post p. For, a post p with $\ell(p) = 0$ and $u(p) = 2$ can be transformed into a post with $\ell(p) = u(p) = 2$ by giving it two dummy edges with zero weight, allowing us to pick the dummy edges in order to make up for the raised lower quota.

The graph $G' = (V', E')$ of the constructed f-factor instance contains the graph $G = (V, E)$ of our WMLQ instance, as shown in Fig. 1. We add a dummy post p_d to V' and connect it to every applicant in A. We connect every post $p_i \in P_1$ to p_d. For every post $p_i \in P \setminus P_1$ we add two dummy vertices q_i^1 and q_i^2 and a triangle on the vertices p_i, q_i^1 and q_i^2. All new edges in $E' \setminus E$ carry zero weight.

We set $f(p_d) = K$, $f(p) = u(p)$ for every $p \in P$ and $f(v) = 1$ for the rest of the vertices. In the initial version of our algorithm, we solve a weighted f-factor problem for every $K \in \{0, 1, ..., |A| + |P_1|\}$, and later we will show a slightly modified version of the f-factor instance so that it is sufficient to construct only two instances.

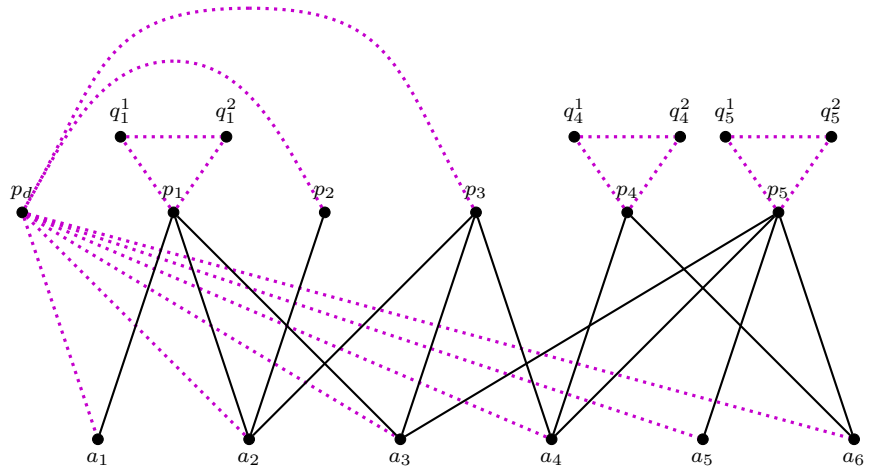

Fig. 1 The transformation from WMLQ to an f-factor problem. The solid edges form G, while the dotted edges are the added ones, carrying weight 0. Here, $P_1 = \{p_2, p_3\}$ and $P \backslash P_1 = \{p_1, p_4, p_5\}$.

First we show that if there is a feasible assignment M in G so that the number of unmatched applicants and the number of closed posts in P_1 add up to K, then it can be extended to an f-factor M' of the same weight in G'. We construct M' starting with M and then adding the following edges to it:

- $\{p_d, a_i\}$ for every applicant a_i that is unmatched in M;
- $\{q_i^1, p_i\}$ and $\{q_i^2, p_i\}$ for every post $p_i \in P \backslash P_1$ that is closed in M;
- $\{q_i^1, q_i^2\}$ for every post $p_i \in P \backslash P_1$ that is open in M;
- $\{p_d, p_i\}$ for every post $p_i \in P_1$ that is closed in M;

For all vertices $v \neq p_d$, it immediately follows from the construction that $\deg_{M'}(v) = f(v)$. The same holds for p_d as well, because an edge is assigned to it either because an applicant is unmatched or because a post in P_1 is closed and we assumed that these add up to K.

It is easy to see that if there is an f-factor M' in G', then its restriction to G is a feasible assignment M of the same weight so that the number of unmatched applicants and the number of closed posts in P_1 add up to K. Since every post $p_i \in P_1$ is connected to p_d and $f(p_i) = 1$, it is either the case that p_i is open in M or $\{p_d, p_i\} \in M'$. Regarding posts outside of P_1, we need to show that the two edges incident to them are either both in G or neither of them are in G. Assume without loss of generality that $\{p_i, q_i^1\} \in M'$ and

$\{p_i, q_i^2\} \notin M'$ for some $p_i \notin P_1$. Since $f(q_i^2) = 1$ and $\deg_{M'}(q_i^2) = 0$, M' cannot be an f-factor.

So far we have shown that it is sufficient to test $|A| + |P_1| + 1$ values for $f(p_d)$, and collect the optimal assignments given by the maximum weight f-factors. Comparing the weight of these locally optimal solutions delivers a global optimum. A slight modification on the the graph corresponding to the f-factor instance will allow us to solve the problem by constructing just two instances, as against $|A| + |P_1| + 1$ instances. Similar to the triangles attached to posts in $P \setminus P_1$, triangles are added to p_d as well. The added vertices have f-value 1 and the added edges carry weight 0. The number of such triangles hanging on p_d is $\left\lceil \frac{|A|+|P_1|}{2} \right\rceil$. These triangles can take up all the f-value of p_d if necessary, but by choosing the edge not incident to p_d they can also allow p_d to fill up its f-value with other edges. Since a triangle either takes up 0 or 2 of p_d's f-value, we need to separate the two different parity cases. Thus, to cover all the $|A| + |P_1| + 1$ cases for possible values for $f(p_d)$, in one instance we set $f(p_d)$ to $|A| + |P_1| + 1$ and in the other instance $f(p_d) = |A| + |P_1|$. □

4 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. Graphs with bounded treewidth occur frequently in real-life instances of various problems, such as expert systems, evolution theory, or natural language processing [5]. More specifically, WMLQ could be used as a potential modeling approach to handle load balancing in telecommunication networks that requires throughput maximization. In WLAN and P2P networks, servers have limited capacities and in addition the loads need to be reasonably balanced among the active servers [3]. This could be quite efficiently modeled using the upper and lower quotas. Moreover, certain P2P and WLAN networks have bounded treewidth [19].

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 $f : \mathbb{R} \to \mathbb{R}$ 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., [27].

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 tree-path connecting them.

The *width* of a tree decomposition with a set of bags $\mathcal{B}$ is $\max_{B\in\mathcal{B}} |B| - 1$. The *treewidth* of a graph G, $\mathrm{tw}(G)$, 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 $\mathrm{tw}(G)$ [6].

Nice tree decomposition. A *nice tree decomposition* is a tree decomposition with the following additional properties: The decomposition tree has a root bag R, and every bag in the tree (including R) is of one of the four types below.

- leaf bag: $|B| = 1$ and B has no child;
- introduce bag: B has exactly one child B_1, so that $B_1 \subset B$ and $|B \setminus B_1| = 1$;
- forget bag: B has exactly one child B_1, so that $B \subset B_1$ and $|B_1 \setminus B| = 1$;
- join bag: B has exactly two children B_1 and B_2, so that $B = B_1 = B_2$.

For every tree decomposition with a specific treewidth, a nice tree decomposition of the same treewidth can be found in linear time [20]. We will henceforth assume we are given such a nice tree decomposition.

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 [32]. 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 $\{1\}$. In contrast with 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 [32] provided a fixed-parameter algorithm for EGCC when parameterized simultaneously by the treewidth of the graph and the maximum domain size, and they showed that the problem is $\mathsf{W}[1]$-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 [32] 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 [32]. The latter makes heavy use of the fact that the domains can be arbitrary sets, whereas in WMLQ, we are confined to intervals.

4.1 Algorithm for bounded treewidth graphs

We will now describe an algorithm for solving WMLQ in polynomial time for graphs with constant treewidth. The algorithm is a dynamic program that inductively computes a set of partial solutions for each bag of the nice tree decomposition and each possible degree profile within that bag. It starts at the leaf bags and then generates partial solutions for each inner node bag by combining or extending partial solutions of its children. The final solution can then be obtained from the root bag. We now formalize the concepts of a partial solution and its corresponding degree profile.

Partial assignments. For any bag B, let $V_B \subseteq V$ denote the set of vertices contained in the union of bags present in the subtree rooted at B. We define the graph $G_B = (V_B, E_B)$ with $E_B := E[V_B] \setminus E[B]$, where $E[U]$ denotes the set of edges with both endpoints in $U \subseteq V$. A *partial assignment* for the subtree rooted at a bag B (or *partial assignment* for B in short) is an assignment $M \subseteq E_B$ of G_B such that $\deg_M(v) = 0$ or $\ell(v) \leq \deg_M(v) \leq u(v)$ for all $v \in V_B \setminus B$.[1] Note that this definition allows applicants and posts in B to have an arbitrary assignment, which is not restricted by their corresponding quota and that by definition of G_B, no vertex in B is assigned to another vertex in B.

Degree profiles. A *degree profile* for bag B is a vector $\alpha \in X_B := \{0, \ldots, u_{\max}\}^B$. We say a partial assignment M for B *agrees* with a degree profile $\alpha \in X_B$, if $\alpha(v) = \deg_M(v)$ for all $v \in B$. For every bag B and every $\alpha \in X_B$, let $\mathcal{M}_B(\alpha)$ be the set of partial assignments for B that agree with α and let

$$W_B(\alpha) := \max\{w(M) : M \in \mathcal{M}_B(\alpha)\} \cup \{-\infty\}$$

denote the optimal value of any assignment that agrees with α for the graph G_B (note that a value of $-\infty$ implies that no partial assignment M agrees with α). We further denote the set of optimal partial assignments agreeing with α by

$$\mathcal{M}^*_B(\alpha) := \{M \in \mathcal{M}_B(\alpha) : w(M) = W_B(\alpha)\}.$$

In the following, we will provide a series of lemmas that reveals how to efficiently obtain an element of $\mathcal{M}^*_B(\alpha)$ for every $\alpha \in X_B$ for a bag B (or showing $\mathcal{M}^*_B(\alpha) = \emptyset$), assuming such representatives for each set $\mathcal{M}^*_{B'}(\alpha')$ have already been computed for every child B' of B and each $\alpha' \in X_{B'}$.

[1] For ease of exposition, we define $\ell(a) := u(a) := 1$ for all $a \in A$. Also recall that $\deg_S(v) := |\delta(v) \cap S|$.

Notation. Before we proceed with the formal statement of the lemmas, we introduce some useful notation. For $U' \subseteq U$ and $\alpha \in \mathbb{Z}^U$ define $\alpha|_{U'}$ as the *restriction* of α to U', i.e., $\alpha|_{U'} \in \mathbb{Z}^{U'}$ and $\alpha|_{U'}(v) = \alpha(v)$ for all $v \in U'$. For $v \in V \setminus U$ and $i \in \mathbb{Z}$ let further $[\alpha, i]_v$ be the *extension* of α to $U \cup \{v\}$ defined by $[\alpha, i]_v(v') := \alpha(v')$ for all $v' \in U$ and $[\alpha, i]_v(v) := i$. For a set of edges S we define the vector $\alpha_{S,U} \in \mathbb{Z}^U$ by $\alpha_{S,U}(v) := \deg_S(v)$ for all $v \in U$.

We first observe that for leaf bags, the only partial assignment possible is the empty assignment.

Lemma 1 *Let $B = \{v\}$ be a leaf bag and let $\alpha \in X_B$. If $\alpha(v) = 0$, then $\mathcal{M}^*_B(\alpha) = \{M_\emptyset\}$, where $M_\emptyset$ is the empty assignment. If $\alpha(v) \neq 0$, then $\mathcal{M}^*_B(\alpha) = \emptyset$.*

Proof This follows directly from the fact that $E_B = \emptyset$ for all leaf bags and thus the only assignment in G_B is the empty assignment. □

For introduce bags, the set of partial assignments is identical to that of its child, extending the degree profiles by a 0-entry for the new vertex.

Lemma 2 *Let B be an introduce bag such that B' is the only child of B and $B \setminus B' = \{v'\}$. Let $\alpha \in X_B$. Then*

$$\mathcal{M}^*_B(\alpha) = \begin{cases} \mathcal{M}^*_{B'}(\alpha|_{B'}) & \text{if } \alpha(v') = 0, \\ \emptyset & \text{otherwise.} \end{cases}$$

Proof Note that $\Gamma(v') \cap V_B \subseteq B$ by Properties 2 and 3 of a tree decomposition. This implies $\delta(v') \cap E_B = \emptyset$ and hence the lemma follows. □

For a forget bag, every partial assignment corresponds to the union of a partial assignment for its child bag and a subset of edges incident to the removed vertex (connecting it to the vertices in forget bag).

Lemma 3 *Let B be a forget bag such that B' is the unique child of B and $B = B' \setminus \{v'\}$ for some $v' \in B'$. Let $\alpha \in X_B$. Let (S^*, i^*) be an optimal solution to*

$$\begin{aligned} \text{[forget]} \qquad \max \quad & w(S) + W_{B'}([\alpha - \alpha_{S,B},\, i - |S|]_{v'}) \\ \text{s.t.} \quad & |S| \leq i, \\ & \alpha_{S,B}(v) \leq \alpha(v) \quad \forall\, v \in B, \\ & S \subseteq \delta(v') \cap \delta(B), \\ & i \in \{0, \ell(v'), \ldots, u(v')\}. \end{aligned}$$

Then $M \cup S^* \in \mathcal{M}^*_B(\alpha)$ for all $M \in \mathcal{M}^*_{B'}([\alpha - \alpha_{S^*,B}, i^* - |S^*|]_{v'})$. If the optimal solution to [forget] has value $-\infty$, then $\mathcal{M}^*_B(\alpha) = \emptyset$.

Proof Assume $\mathcal{M}_B(\alpha) \neq \emptyset$ and let $M' \in \mathcal{M}^*_B(\alpha)$. Let $S' := M' \cap \delta(v') \cap \delta(B)$ and let $i' := \deg_{M'}(v')$. Observe that (S', i') is a feasible solution to [forget] and that $M' \setminus S' \in \mathcal{M}_{B'}([\alpha - \alpha_{S',B},\, i' - |S'|]_{v'})$. We conclude that

$$\begin{aligned} w(M') &\leq w(S') + W_{B'}([\alpha - \alpha_{S',B}, i' - |S'|]_{v'}) \\ &\leq w(S^*) + W_{B'}([\alpha - \alpha_{S^*,B},\, i^* - |S^*|]_{v'}). \end{aligned}$$

In particular, this implies that the optimal solution value of [forget] is finite and thus there is some $M \in \mathcal{M}^*_{B'}([\alpha - \alpha_{S^*,B},\, i^* - |S^*|]_{v'})$.

Thus let $M^* := M \cup S^*$. Observe that indeed $\deg_{M^*}(v) = \deg_M(v) + \deg_{S^*}(v) = \alpha(v) - \alpha_{S^*,B}(v) + \alpha_{S^*,B}(v) = \alpha(v)$ for all $v \in B$. Furthermore $\deg_{M^*}(v) = \deg_M(v) \in \{0, \ell(v), \ldots, u(v)\}$ for all $v \in V_B \setminus B'$ by feasibility of M. Finally, $\deg_{M^*}(v') = i^* \in \{0, \ell(v'), \ldots, u(v')\}$, implying $M^* \in \mathcal{M}_B(\alpha)$. As $w(M^*) = w(S^*) + W_{B'}([\alpha - \alpha_{S^*},\, i^* - |S^*|]_{v'}) \geq w(M')$, we conclude that indeed $M^* \in \mathcal{M}^*_B(\alpha)$. □

For join bags, every partial assignment is a union of partial assignments for the two child bags.

Lemma 4 *Let B be a join bag such that $B = B_1 = B_2$ for the two children B_1, B_2 of B. Let $\alpha \in X_B$. Let (α^*_1, α^*_2) be an optimal solution to*

$$\begin{aligned} \text{[join]} \qquad \max \;& W_{B_1}(\alpha_1) + W_{B_2}(\alpha_2) & \\ \text{s.t.} \;& \alpha_1(v) + \alpha_2(v) = \alpha(v) & \forall\, v \in B, \\ & \alpha_1 \in X_{B_1},\; \alpha_2 \in X_{B_2}. & \end{aligned}$$

Then $M_1 \cup M_2 \in \mathcal{M}^*_B(\alpha)$ for all $M_1 \in \mathcal{M}^*_{B_1}(\alpha^*_1),\; M_2 \in \mathcal{M}^*_{B_2}(\alpha^*_2)$. If the optimal solution to [join] has value $-\infty$, then $\mathcal{M}^*_B(\alpha) = \emptyset$.

Proof Let $M^* := M_1 \cup M_2$ for some $M_1 \in \mathcal{M}^*_{B_1}(\alpha^*_1),\; M_2 \in \mathcal{M}^*_{B_2}(\alpha^*_2)$. We first observe that $V_{B_1} \cap V_{B_2} = B$ by Properties 2 and 3 of the tree decomposition and hence $M_1 \cap M_2 = \emptyset$. This implies that

$$\deg_{M^*}(v) = \begin{cases} \deg_{M_1}(v) \in \{0, \ell(v), \ldots, u(v)\} & \text{if } v \in V_{B_1} \setminus B, \\ \deg_{M_2}(v) \in \{0, \ell(v), \ldots, u(v)\} & \text{if } v \in V_{B_2} \setminus B, \\ \deg_{M_1}(v) + \deg_{M_2}(v) = \alpha(v) & \text{if } v \in B. \end{cases}$$

Hence $M^* \in \mathcal{M}_B(\alpha)$.

Now let $M' \in \mathcal{M}_B(\alpha)$. Let $M'_1 := M' \cap E_{B_1}$ and $M'_2 := M' \cap E_{B_2}$. We observe that $(\alpha_{M_1,B_1}, \alpha_{M_2,B_2})$ is a feasible solution to [join] and hence $w(M') = w(M'_1) + w(M'_2) \leq w(M_1) + w(M_2) = w(M^*)$. □

Finally, we observe that after computing $W_R(\alpha)$ and the corresponding elements of $\mathcal{M}^*_R(\alpha)$ for each α for the root bag R, an optimal assignment for G can be easily obtained.

Lemma 5 *Let (S^*, α^*) be an optimal solution to*

$$\begin{aligned} [\text{root}] \qquad \max \quad & W_R(\alpha) + w(S) \\ \text{s.t.} \quad & \alpha(v) + \deg_S(v) \in \{0, \ell(v), \dots, u(v)\} \quad \forall\, v \in R, \\ & \alpha \in X_R,\ S \subseteq E[R]. \end{aligned}$$

Then $S^* \cup M$ is an optimal solution to WMLQ *for any $M \in \mathcal{M}_R^*(\alpha^*)$.*

Proof Let $M^* := S^* \cup M$ for some $M \in \mathcal{M}_R^*(\alpha^*)$. Note that for $v \in V \setminus R$, we have $S^* \cap \delta(v) = \emptyset$ and hence $\deg_{M^*}(v) = \deg_M(v) \in \{0, \ell(v), \dots, u(v)\}$ by the feasibility of M. Furthermore, for $v \in R$, we have $\deg_{M^*}(v) = \alpha^*(v) + \deg_{S^*}(v) \in \{0, \ell(v), \dots, u(v)\}$ by the feasibility of S^* for [root]. We conclude that M^* is indeed a feasible solution to WMLQ.

Now let $M' \subseteq E$ be any solution to WMLQ. Define $S' := M' \cap E[R]$ and $\alpha' := \alpha_{M',R} - \alpha_{S',R}$. Observe that (S', α') is a feasible solution to [root] and that further $M' \setminus S' \in \mathcal{M}_R(\alpha')$. We conclude that

$$w(M') \leq W_R(\alpha') + w(S') \leq W_R(\alpha^*) + w(S^*) = w(M^*),$$

and thus M^* is indeed an optimal solution to WMLQ. □

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

Proof In order to solve a given WMLQ instance, the algorithm starts by computing a nice tree decomposition of G of width $\mathrm{tw}(G)$. Note that T is of the same order for tree decompositions and nice tree decompositions. Using Lemmas 1 to 5, we can inductively compute a representative $M \in \mathcal{M}_B^*(\alpha)$ for every bag B and every $\alpha \in X_b$, or deduce that $\mathcal{M}_B^*(\alpha) = \emptyset$. We first observe that $|X_B| = (u_{\max}+1)^{|B|}$, thus only $(u_{\max}+1)^{\mathrm{tw}(G)+1}$ representatives have to be computed per bag. Furthermore, for each of the above lemmas, the necessary computations to derive an $M \in \mathcal{M}_B^*(\alpha)$ from representatives of $\mathcal{M}_{B'}^*(\alpha')$ of children B' of B can be done in time $O((u_{\max})^{2\,\mathrm{tw}(G)+2})$. This is obvious for Lemmas 1 and 2. For Lemmas 3 to 5 we observe that the sets of feasible solutions for the corresponding optimization problems [forget], [join], and [root] have size at most $2^{|B|} \cdot (u_{\max}+1)$, $(u_{\max}+1)^{2|B|}$, and $2^{|R|^2} \cdot (u_{\max})^{|R|}$, respectively (note that without loss of generality we can assume $|R|$ to be of constant size by introducing at most $\mathrm{tw}(G)$ additional forget bags). The theorem then follows from the fact that the number of bags is linear. □

4.2 W[1]-hardness for parameterizing by treewidth only

While our algorithm runs in polynomial time for bounded treewidth, the degree of the polynomial depends on the treewidth and 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 FPT $\neq$ W[1].

Problem 2 Minimum Maximum Outdegree
Input: A graph $G = (V, E)$, edge weights $w : E \to \mathbb{Z}_+$ encoded in unary and 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 [34]) Minimum Maximum Outdegree *is W[1]-hard when parameterized by treewidth.*

Theorem 6 mlq *is* 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 $p_v \in P$ with lower quota $\ell(p_v) = 0$ and upper quota $u(p_v) = r$.
- For every edge $e = \{v, v'\} \in E$, we introduce two posts $p_{e,v}$ and $p_{e,v'}$ with identical lower and upper quotas of $w(e) + 1$, i. e. ,

$$\ell(p_{e,v}) = \ell(p_{e,v'}) = u(p_{e,v}) = u(p_{e,v'}) = w(e) + 1.$$

  We also add $2w(e) + 1$ applicants $a^1_{e,v}, \ldots, a^{w(e)}_{e,v}, a^1_{e,v'}, \ldots, a^{w(e)}_{e,v'}, z_e$, which are connected to the posts by the edges

$$\{p_v, a^i_{e,v}\}, \{a^i_{e,v}, p_{e,v}\}, \{p_{v'}, a^i_{e,v'}\}, \{a^i_{e,v'}, p_{e,v'}\} \text{ for } i \in \{1, \ldots, w(e)\}$$

  as well as $\{p_{e,v}, z_e\}$ and $\{z_e, p_{e,v'}\}$. This construction is shown in Fig. 2.

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 weighted outdegree.

First assume there is an orientation D of G with maximum weighted outdegree at most r. Then consider the assignment that assigns for every oriented edge $(v, v') \in D$ the $w(e)$ applicants $a^i_{e,v}$ to p_v and the $w(e) + 1$ applicants $a^i_{e,v'}$ 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

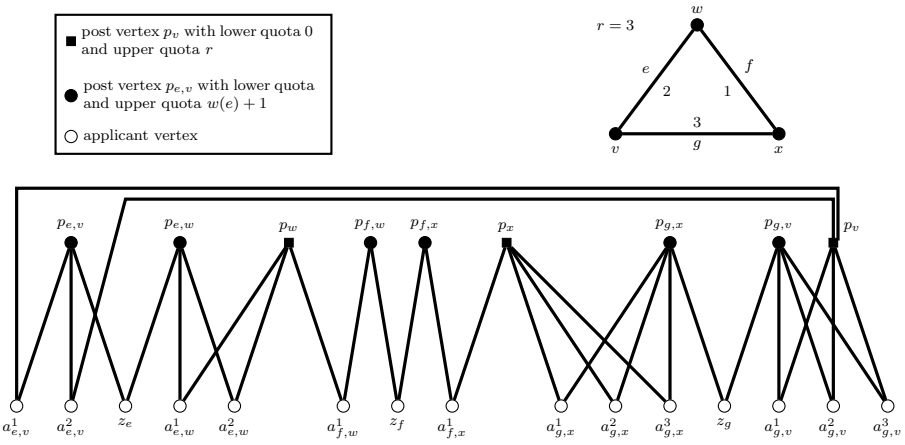

Fig. 2 The transformation of the MINIMUM MAXIMUM OUTDEGREE instance in the upper right corner to a MLQ instance. The numbers on the edges of the MINIMUM MAXIMUM OUTDEGREE instance are the edge weights.

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 $a^i_{e,v'}$ are assigned to $p_{v'}$ and none of the applicants $a^i_{e,v}$ 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 $\max\{\mathrm{tw}(G), 3\}$. 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 $B_{z_e} = \{p_{e,v}, p_{e,v'}, z_e\}$, $B_{a^i_{e,v}} = \{p_v, p_{e,v}, a^i_{e,v}\}$ and $B_{a^i_{e,v'}} = \{p_{v'}, p_{e,v'}, a^i_{e,v'}\}$ for $i \in \{1, \ldots, w(e)\}$ 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 $\max\{\mathrm{tw}(G), 3\}$. □

5 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 described formally 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 (p', A') 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 ∈ P : |Γ(p)| ≥ ℓ(p)}.
Initialize A_0 = A.
while P_0 ≠ ∅ do
    Find a pair p' ∈ P_0 and A' ⊆ Γ(p') ∩ A_0 with |A'| ≤ u(p') such that Σ_{a∈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 ∈ P_0 with ℓ(p) > |Γ(p) ∩ A_0| do
        Remove p from P_0.
    end for
end while
```

Remark 2 As an alternative to Algorithm 1, one could use a reduction from WMLQ to the set packing problem. 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. A feasible set packing then corresponds to a feasible assignment of the same weight. However, if the difference between p's upper and lower quota is not bounded by a constant, this would create an exponential-sized input for the set packing problem and we could only employ an oracle-based algorithm known for the set packing problem to solve WMLQ. The greedy algorithm known for the set packing problem [7] can be made to work in a fashion similar to the algorithm presented above.

In the following we give a tight analysis of the algorithm, establishing approximation guarantees in terms of the number of posts $|P|$, the 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 that the approximation ratios given above for WMLQ are almost tight from the point of view of complexity theory.

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 $O(|E| \log |E|)$.*

Proof Let p'_i be the post chosen by the algorithm in iteration i and let A'_i be the corresponding set of applicants for every $i \in \{1, \ldots, n\}$. Furthermore,

consider an optimal solution of weight OPT, consisting of open posts $p_1, \ldots, p_k$ and the corresponding sets of applicants $A_1, \ldots, A_k$ assigned to those posts.

We first observe that the first two approximation ratios of $|P|$ and $|A|$ are already achieved by the initial selection of p'_1 and A'_1 chosen in the first round of the algorithm. For every $i \in \{1, \ldots, k\}$, post p_i is an admissible post in the first iteration of the algorithm. The first iteration's choice of the pair (p'_1, A'_1) implies $\sum_{a \in A'_1} w(a, p'_1) \geq \sum_{a \in A_i} w(a, p_i) \geq w(a', p_i)$ for every $a' \in A_i$. As the optimal solution opens at most $|P|$ posts and serves at most $|A|$ applicants, we deduce that $\min\{|P|, |A|\} \cdot \sum_{a \in A'_1} w(a, p'_1) \geq \text{OPT}$.

We now turn our attention to the remaining approximation guarantees, which are $u_{\max} + 1$ for WMLQ and $\sqrt{|A|} + 1$ for MLQ. For every $i \in \{1, \ldots, k\}$, let $\pi(i)$ denote the first iteration of the algorithm such that $A'_{\pi(i)} \cap A_i \neq \emptyset$ or $p'_{\pi(i)} = p_i$. This is the first iteration in which post p_i is opened or an applicant assigned to it in the optimal solution becomes assigned. Note that such an iteration exists, because p_i is not admissible after the termination of the algorithm. Furthermore, observe that $\sum_{a \in A'_{\pi(i)}} w(a, p'_{\pi(i)}) \geq \sum_{a \in A_i} w(a, p_i)$, because the pair (p_i, A_i) was a valid choice for the algorithm in iteration $\pi(i)$. Now for iteration j define $P_j := \{i : \pi(i) = j\}$ and observe that $|P_j| \leq |A'_j| + 1$, because P_j can only contain one index i' with $p_{i'} = p'_j$, and all other $i \in P_j \setminus \{i'\}$ must have $A_i \cap A'_j \neq \emptyset$ (where the sets A_i are disjoint). We conclude that

$$\begin{aligned} \text{OPT} &= \sum_{i=1}^{k} \sum_{a \in A_i} w(a, p_i) \leq \sum_{i=1}^{k} \sum_{a \in A'_{\pi(i)}} w(a, p'_{\pi(i)}) \\ &\leq \sum_{j=1}^{n} |P_j| \sum_{a \in A'_j} w(a, p'_j) \leq \sum_{j=1}^{n} (|A'_j| + 1) \sum_{a \in A'_j} w(a, p'_j). \end{aligned}$$

Note that $|A'_j| \leq u_{\max}$ and therefore

$$\text{OPT} \leq (u_{\max} + 1) \sum_{j=1}^{n} \sum_{a \in A'_j} w(a, p'_j),$$

proving the third approximation guarantee. Now consider the unit-weight MLQ case and define $A' = \bigcup_{j=1}^{n} A'_j$. If $|A'| \geq \sqrt{|A|}$, then $\sqrt{|A|}|A'| \geq |A| \geq \text{OPT}$. Therefore assume $|A'| < \sqrt{|A|}$. Note that in this case, the above inequalities imply $\text{OPT} \leq (|A'| + 1)|A'| \leq (\sqrt{|A|} + 1)|A'|$, proving the improved approximation guarantee for MLQ.

We now turn to proving the bound on the running time. We will describe how to implement the search for the greedy choice of the pair (p', A') in each iteration efficiently using a heap data structure. Initially, for every post p, we sort the applicants in its neighborhood by non-increasing order of $w(a, p)$. This takes time at most $O(|E| \log |E|)$ as the total number of entries to sort is $\sum_{p \in P} |\Gamma(p)| = |E|$. We then introduce a heap containing all admissible posts, and associate with each post p the total weight of the first $u(p)$ edges

in its neighborhood list. Note that these entries can be easily kept up to date whenever the algorithm opens a post and assigns applicants to it: In the list of every other post p we simply replace the assigned applicants with the first not-yet-assigned entry in the list (or we remove the post if less than $\ell(p)$ applicants are available). As every edge in the graph can only trigger one such replacement, only $O(|E|)$ updates can occur and each of these requires $O(\log |P|)$ time for reinserting the post at the proper place in the heap. Now, in each iteration of the algorithm, the optimal pair (p', A') can be found by retrieving the maximum element from the heap. This happens at most $|P|$ times and requires $O(\log |P|)$ time in each step. □

Example 8 *The following two examples show that our analysis of the greedy algorithm is (asymptotically) tight for each of the described approximation factors.*

(a) The bounds $|P|$ and $u_{\max} + 1$ are tight, and $\sqrt{|A|} + 1$ is asymptotically tight:
Consider an instance of MLQ *with $k+1$ posts $p_0, \ldots, p_k$ and $k(k+1)$ applicants $a_{0,1}, \ldots, a_{0,k}, a_{1,1}, \ldots, a_{k,k}$. Let $\ell(p_i) = u(p_i) = k$ for $i \in \{0, \ldots, k\}$. Each applicant $a_{i,j}$ applies to post i, and if $i > 0$, additionally to post 0. For the greedy algorithm, opening post p_0 and assigning applicants $a_{1,1}, \ldots, a_{k,k}$ to it is a valid choice in its first iteration, after which no further posts are admissible. Thus, it only assigns k applicants in total. The optimal solution, however, can assign all $k(k+1)$ applicants by assigning applicants $a_{i,1}, \ldots, a_{i,k}$ to p_i for each i. Therefore, the greedy algorithm cannot achieve an approximation factor better than $k+1$ on this family of instances, for which $|P| = k+1$, $\sqrt{|A|} < k+1$, and $u_{\max} = k$.*

(b) The bound $|A|$ is tight:
To see that the approximation ratio of $|A|$ is tight for WMLQ *consider the following instance with k posts $p_1, \ldots, p_k$ and k applicants $a_1, \ldots, a_k$. Let $\ell(p_i) = 0$ and $u(p_i) = k$ for every i. Every applicant applies for every post, and $w(a_i, p_i) = 1$ for every i but $w(a_i, p_j) = \varepsilon$ for every $j \neq i$ for some arbitrarily small $\varepsilon > 0$. In its first iteration, the greedy algorithm might choose to open post p_1 and assign all applicants to it. This solution accumulates a weight of $1+(k-1)\varepsilon$, while the weight of the optimal solution is $k = |A|$.*

Theorem 9 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 $\mathsf{P} = \mathsf{NP}$, even when restricting to instances where $\ell(p) = u(p)$ for every $p \in P$ and $|\Gamma(a)| \leq 2$ for every $a \in A$.*

Proof Once again we use the maximum independent vertex set problem. Given an instance of MIS on a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, we create an MLQ instance with n posts $p_1, \ldots, p_n$, post p_i corresponding to vertex v_i. We also introduce $n^2 - m$ applicants as follows. Initially, we introduce n applicants $a_{i,1}, a_{i,2}, ..., a_{i,n}$ applying for each post p_i. Then, for every edge $\{v_i, v_j\} \in E$,

we merge the applicants $a_{i,j}$ and $a_{j,i}$, obtaining a single applicant applying for both p_i and p_j. Furthermore, we set $\ell(p_j) = u(p_j) = n$ for every post. This construction is shown in Fig. 3.

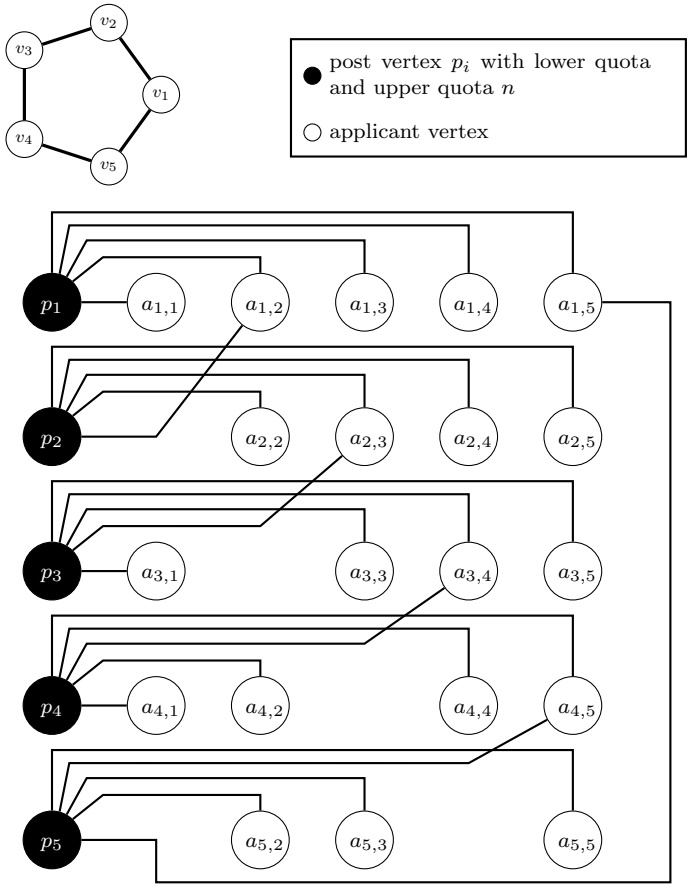

Fig. 3 The transformation of the MIS instance in the upper left corner to a MLQ instance.

Note that due to the choice of upper and lower bounds, any open post must be assigned to all the applicants in its neighborhood. Thus, a solution to the MLQ instance is feasible if and only if $\Gamma(p_i) \cap \Gamma(p_j) = \emptyset$ for all open posts p_i and p_j with $i \neq j$, which is equivalent to v_i and v_j not being adjacent in G by construction of the instance. Therefore, the MLQ instance has a feasible solution opening k posts (and thus serving kn applicants) if and only if there is an independent set of size k in G. We conclude that $\mathrm{OPT}_{\mathrm{MLQ}} = n \cdot \mathrm{OPT}_{\mathrm{MIS}}$ for the two instances under consideration.

Note that in the constructed MLQ instance, $n = |P| = u_{\max} \geq \sqrt{|A|}$. Therefore any approximation algorithm with a factor better than $|P|^{1-\varepsilon}$ or $\sqrt{|A|}^{1-\varepsilon}$ or $u_{\max}^{1-\varepsilon}$ for $\varepsilon > 0$ yields a solution of the instance that serves at

least $(1/n^{1-\varepsilon})\mathrm{OPT}_{\text{MLQ}}$ applicants and therefore opens at least $(1/n^{2-\varepsilon})\mathrm{OPT}_{\text{MLQ}} = (1/n^{1-\varepsilon})\mathrm{OPT}_{\text{MIS}}$ posts, corresponding to an independent set of the same size. By [35], this implies $\mathsf{P} = \mathsf{NP}$. □

6 Matchings with lower quotas in general graphs

Throughout this paper, we focused on many-to-one matchings in bipartite graphs because these fit most applications in the centralized formation of groups that motivated our investigation. A straightforward generalization of WMLQ to matchings in an arbitrary (not necessarily bipartite) graph G allows *all* vertices of the graph to have lower and upper quotas.

Problem 3 GWMLQ
Input: $\mathcal{I} = (G, w, \ell, u)$*; a not necessarily bipartite graph* $G = (V, E)$ *with edge weights* w*, lower quotas* ℓ *and upper quotas* u*.*
Task: Find an assignment of maximum weight.
If $w(e) = 1$ *for all* $e \in E$*, we refer to the problem as* GMLQ*.*

One can see this generalization as a variant of the D-matching problem (see Section 1.1), where each vertex has a domain consisting of 0 and an interval. Clearly, the hardness results derived in the previous sections are valid for GWMLQ as well. We now briefly argue that the positive results from Sections 3 and 4 carry over to this generalized setting. However, our approximation results do not hold even if G is bipartite and only a single applicant is equipped with lower and upper quotas. In fact, GWMLQ does not allow for any approximation even in this very restricted case unless $\mathsf{P} = \mathsf{NP}$.

The two positive results in Section 3, namely Theorems 2 and 3, are applicable to GWMLQ. Note that Theorem 2 (bounded degree for all posts) is a special case of Theorem 3 (bounded upper quota for all posts).

Theorem 10 GWMLQ *can be solved in polynomial time when restricted to instances with* $u(v) \leq 2$ *for all* $v \in V$*.*

Proof We will work with the proof of Theorem 3, which requires some simple modifications to fit the case of arbitrary graphs. All we need to do is to add a dummy vertex v_d to G – this resembles dummy post p_d in the proof of Theorem 3. The steps corresponding to a post vertex should now be executed for all vertices of the graph. We can assume there are no vertices with lower quota 0 and upper quota 2 by a similar reasoning given in Theorem 2. For every vertex v_i with $\ell(v_i) = 2$, we add two dummy vertices q_i^1 and q_i^2 and connect them to each other and v_i. For all of these vertices, we set $\ell(q_i^1) = u(q_i^1) = \ell(q_i^2) = u(q_i^2) = 1$. Then, the dummy vertex v_d is connected to vertices with upper quota 1. We finish the construction by adding triangles to v_d to ensure that only two f-factors need to be computed. The arguments in the proof of Theorem 3 can now be applied to this f-factor instance. □

As for Theorem 4, the algorithm for bounded treewidth and upper quota carries over to GWMLQ without any modification. Note that in the proof we never used the bipartiteness of G or that $u(a) = 1$ for the applicants.

Theorem 11 GWMLQ *can be solved in time* $O(T + (u_{\max})^{3\,\mathrm{tw}(G)+3}|E|)$*, where* T *is the time needed for computing a tree decomposition of* G *of width* $\mathrm{tw}(G)$*. In particular,* GWMLQ *can be solved in polynomial time when restricted to instances of bounded treewidth, and* WMLQ *parameterized by* $\max\{\mathrm{tw}(G), u_{\max}\}$ *is fixed-parameter tractable.*

Finally, we prove that Algorithm 1 cannot be generalized even for bipartite MLQ with lower and upper quotas on both sides.

Theorem 12 *It is* NP*-hard to decide whether* $\mathrm{OPT} > 0$ *for an instance of* GMLQ*, even if the graph is bipartite and on one side of the bipartition all vertices except for one have unitary upper and lower quota.*

Proof To every instance of MIS we construct an instance of GMLQ so that the MIS instance admits an independent set of size K if and only if $\mathrm{OPT} > 0$ for the GMLQ instance. We start with the same MLQ instance that was constructed from an MIS instance in the proof of Theorem 9. The changes are depicted in Fig. 4. A dummy applicant a_d is added to the graph and connected to all posts. We set $\ell(a_d) = u(a_d) = K$ and change $\ell(p) = u(p)$ to $n + 1$ for every post $p \in P$.

Since every post is adjacent to exactly $n + 1$ applicants, opening a post requires allocating all its applicants to it, including a_d as well. Thus, opening any post implies allocating a_d to exactly K posts. These K open posts do not share applicants other than a_d, which is equivalent to the K vertices corresponding to them in the MIS instance forming an independent set. □

7 Conclusion

We discussed the complexity, approximability and fixed-parameter tractability of WMLQ from various viewpoints such as bounded degree, quota and treewidth.

Further work on the topic might include imposing common quotas on some groups of posts. That is, we may have subsets $P_1, \ldots, P_k$, where for each i $(1 \leq i \leq k)$, $P_i \subseteq P$, P_i has a *common quota* $u(P_i) \geq 1$, where $u(P_i) \leq \sum_{p \in P_i} u(p)$, and any assignment M must now satisfy the additional property that $\sum_{p \in P_i} |\delta(p) \cap M| \leq u(P_i)$. Common quotas are similar to regional caps studied in Economics [17] and they can model constraints such as the limited availability of resources required for certain projects – for example P_1 might correspond to those projects that require access to high-performance computing facilities.

We have seen that WMLQ as defined in Problem 1 has a natural application in the context of student-project allocation, where the weight on a given

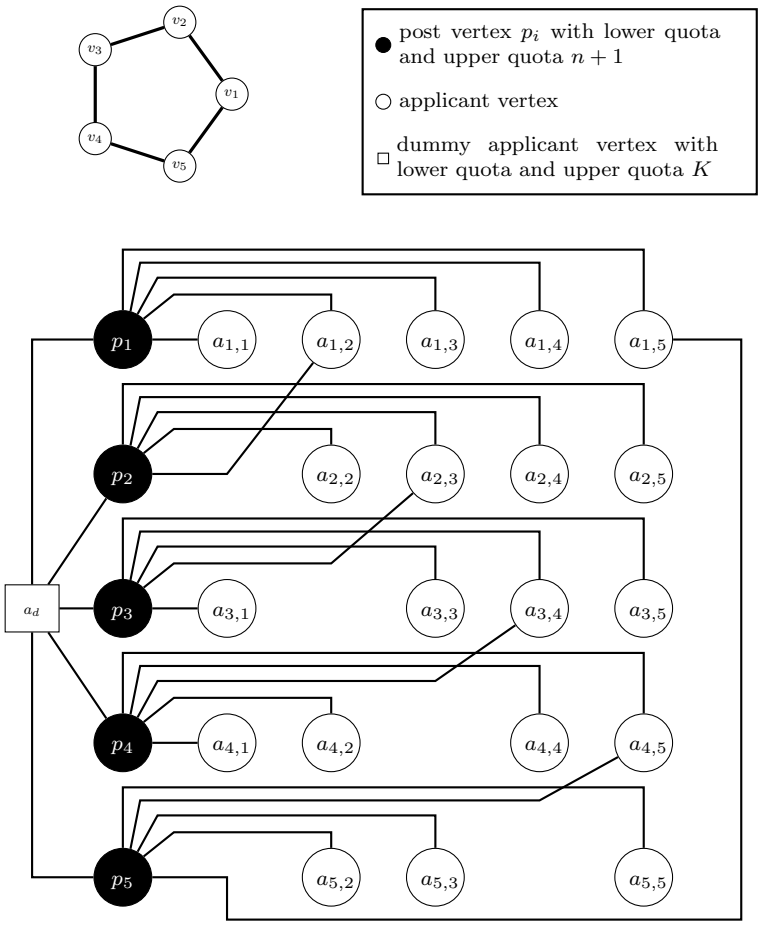

Fig. 4 The transformation of the MIS instance in the upper left corner to a generalized MLQ instance.

edge (s, p) corresponds to the utility of student s being assigned to project p. However in many applications students have ordinal preferences over projects. Cardinal utilities can of course follow from these via the use of Borda scores, so we can obtain WMLQ as before. But ordinal preferences themselves allow alternative optimality criteria to be formulated. For example we may optimize on the *profile* of a matching M, which is a vector whose ith position indicates the number of students who obtain their ith-choice project in M [24]. A *greedy maximum matching* is a matching whose profile is lexicographically maximum, taken over all maximum cardinality matchings, whilst a *generous maximum matching* is a matching whose reverse profile is lexicographically minimum, taken over all maximum cardinality matchings. There are efficient algorithms to find greedy and generous maximum matchings in the absence

of lower quotas [21], but it remains open to extend the positive results in this paper to the setting involving both lower quotas and preferences.

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

References

1. D. J. Abraham, R. W. Irving, and D. F. Manlove. Two algorithms for the Student-Project Allocation problem. *Journal of Discrete Algorithms*, 5(1):79–91, 2007.
2. P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1-2):123–134, 2000.
3. Y. Bejerano and S. J. Han. Cell breathing techniques for load balancing in wireless lans. *IEEE Transactions on Mobile Computing*, 8(6):735–749, 2009.
4. P. Biró, T. Fleiner, R. W. Irving, and D. F. Manlove. The College Admissions problem with lower and common quotas. *Theoretical Computer Science*, 411:3136–3153, 2010.
5. H. L. Bodlaender. A tourist guide through treewidth. *Acta cybernetica*, 11(1-2):1, 1994.
6. H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.
7. B. Chandra and M. M. Halldórsson. Greedy local improvement and weighted set packing approximation. *Journal of Algorithms*, 39(2):223–240, 2001.
8. G. Cornuéjols. General factors of graphs. *Journal of Combinatorial Theory, Series B*, 45(2):185–198, 1988.
9. D. Fragiadakis, A. Iwasaki, P. Troyan, S. Ueda, and M. Yokoo. Strategyproof matching with minimum quotas. *ACM Transactions on Economics and Computation*, 4(1):6:1–40, Jan 2016.
10. H. N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of STOC '83: the 15th Annual ACM Symposium on Theory of Computing*, pages 448–456. ACM, 1983.
11. H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of SODA '90: the 1st ACM-SIAM Symposium on Discrete Algorithms*, pages 434–443. ACM-SIAM, 1990.
12. M. Goto, N. Hashimoto, A. Iwasaki, Y. Kawasaki, S. Ueda, Y. Yasuda, and M. Yokoo. Strategy-proof matching with regional minimum quotas. In *Proceedings of the AAMAS '14: the 13th International Conference on Autonomous Agents and Multi-agent Systems*, pages 1225–1232. IFAAMAS, 2014.

13. M. Goto, R. Kurata, N. Hamada, A. Iwasaki, and M. Yokoo. Improving fairness in nonwasteful matching with hierarchical regional minimum quotas. In *Proceedings of AAMAS '15: the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 1887–1888. IFAAMAS, 2015.
14. K. Hamada, K. Iwama, and S. Miyazaki. The hospitals/residents problem with lower quotas. *Algorithmica*, 74(1):440–465, 2014.
15. J. P. Hanna, M. Albert, D. Chen, and P. Stone. Minimum cost matching for autonomous carsharing. *IFAC-PapersOnLine*, 49(15):254–259, 2016. 9th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2016.
16. K. Iwama, S. Miyazaki, and H. Yanagisawa. Improved approximation bounds for the student-project allocation problem with preferences over projects. *Journal of Discrete Algorithms*, 13:59–66, 2012.
17. Y. Kamada and F. Kojima. Stability and strategy-proofness for matching with constraints: A problem in the japanese medical match and its solution. *American Economic Review*, 102(3):366–70, May 2012.
18. N. Kamiyama. A note on the serial dictatorship with project closures. *Operations Research Letters*, 41:559–561, 2013.
19. R. Klasing, A. Kosowski, and A. Navarra. Cost minimization in wireless networks with a bounded and unbounded number of interfaces. *Networks*, 53(3):266–275, 2009.
20. T. Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.
21. A. Kwanashie, R. W. Irving, D. F. Manlove, and C. T. S. Sng. Profile-based optimal matchings in the Student/Project Allocation problem. In *Proceedings of IWOCA '14: the 25th International Workshop on Combinatorial Algorithms*, volume 8986 of *Lecture Notes in Computer Science*, pages 213–225. Springer, 2015.
22. L. Lovász. On the structure of factorizable graphs. *Acta Mathematica Academiae Scientiarum Hungaricae*, 23(1-2):179–195, 1972.
23. L. Lovász. Antifactors of graphs. *Periodica Mathematica Hungarica*, 4(2-3):121–123, 1973.
24. D. F. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2013.
25. D. F. Manlove and G. O'Malley. Student project allocation with preferences over projects. *Journal of Discrete Algorithms*, 6:553–560, 2008.
26. D. Monte and N. Tumennasan. Matching with quorums. *Economics Letters*, 120:14–17, 2013.
27. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
28. S. Ólafsson. Weighted matching in chess tournaments. *The Journal of the Operational Research Society*, 41(1):17–24, 1990.
29. M. D. Plummer. Graph factors and factorization: 1985–2003: A survey. *Discrete Mathematics*, 307(7-8):791–821, 2007.
30. M. E. Renda and U. Straccia. Web metasearch: Rank vs. score based rank aggregation methods. In *Proceedings of the 2003 ACM Symposium on*

Applied Computing, SAC '03, pages 841–846, New York, NY, USA, 2003. ACM.
31. S. Röllin and O. Schenk. Maximum-weighted matching strategies and the application to symmetric indefinite systems. *Applied Parallel Computing. State of the Art in Scientific Computing*, pages 808–817, 2006.
32. M. Samer and S. Szeider. Tractable cases of the extended global cardinality constraint. *Constraints*, 16:1–24, 2011.
33. A. Sebő. General antifactors of graphs. *Journal of Combinatorial Theory, Series B*, 58(2):174–184, 1993.
34. S. Szeider. Not so easy problems for tree decomposable graphs. *CoRR*, abs/1107.1177, 2011.
35. D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007.