European Journal of Operational Research 307 (2023) 392–405

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Online voluntary mentoring: Optimising the assignment of students and mentors

Péter Biró[a,b,*], Márton Gyetvai[a]

[a] Institute of Economics, CERS, Tóth Kálmán u. 4, Budapest, 1097, Hungary
[b] Corvinus University of Budapest, Fővám tér 8, Budapest, 1093, Hungary

ARTICLE INFO

Article history:
Received 15 March 2021
Accepted 5 August 2022
Available online 10 August 2022

Keywords:
Assignment
Resource allocation
Matching under preferences
Integer programming
Computational simulation

ABSTRACT

After the closure of the schools in Hungary from March 2020 due to the pandemic, many students were left at home with no or not enough parental help for studying, and in the meantime some people had more free time and willingness to help others in need during the lockdown. In this paper we describe the optimisation aspects of a joint NGO project for allocating voluntary mentors to students using a web-based coordination mechanism. The goal of the project has been to form optimal pairs and study groups by taking into account the preferences and the constraints of the participants. In this paper, we present the optimisation concept and the integer programming techniques used for solving the allocation problems. Furthermore, we conducted computational simulations on real and generated data to evaluate the performance of this dynamic matching scheme under different parameter settings.

1. Introduction

In Hungary, after the lockdown due to the spreading of the COVID-19 virus, the government announced the closure of all schools on 13 March 2020 (Friday evening) effective from 16 March, and they requested all schools (both primary and secondary schools with students of ages between 6 and 18) to start online/distance education immediately. There was no central recommendation about the technology and methodology to be used, so this was decided mainly by the board of each school selecting from a wide range of online platforms (e.g. Google Classroom, MS Teams, etc.) or just sending the weekly assignments by mail.

A large number of students had difficulties to adapt to the distance education, partly because of missing equipment or internet connection, but also because the parents were not able to help them at home, due to lacking the knowledge in special subjects or just because of being at work (e.g., including those parents working intensively in health care). At the same time, many people, especially the elderly had to stay at home with spare time. Furthermore, in Hungary there is also a programme for secondary school students for voluntary work which they can get credit for (that counts extra points at the centralised university admission), and very few possibilities remained for such services under the strict social distancing rules imposed. Therefore, there was need for mentoring, and also a significant amount of potential mentors, old and young alike.

A project was proposed in early April and then officially started with the opening of a web-application (`onkentesmentoralas.hu`) in early May in the cooperation of three parties. On behalf of the Institute of Economics of CERS, the Mechanism Design and the Education Economics research groups offered help in designing a mechanism for allocating students to mentors. The Hungarian Reformed Church Aid is a humanitarian organisation which had a link to a governmental action group devoted to coordinate the voluntary help. The third party in the project was #school, a private company providing an online teaching platform that went into operation with 100k registered users soon after the online education started. The latter two parties had had a related collaboration in the past, where they organised the mentoring of seriously disadvantaged children.

The design approach was rather complex, taking into account the preferences of both sides, and also allowing the formation of study groups, besides the mentor-student pairs. The basic requirement of creating a pair is to have a subject (e.g., Maths at year 7) that is both requested by a student and offered by a mentor.

* Corresponding author at: Institute of Economics, CERS, Tóth Kálmán u. 4, Budapest 1097, Hungary.
E-mail address: birop@econ.core.hu (P. Biró).

https://doi.org/10.1016/j.ejor.2022.08.008

The students can request several subjects listing them according to their preferences and whether they are willing to study in groups or only in pairs. The mentors can specify their preferences over the subjects they offer to teach, whether they are willing to supervise groups (or only individuals), and they can also set preferences over some characteristics of the students, such as age, performance level in that subject measured by grades, and social status (i.e. whether the mentor prefers to teach seriously disadvantaged students).

In the optimisation we carefully considered several potential goals and implemented a combination of them, such as maximising the number of students matched, the overall volume of teaching hours, the preferences of both the students and mentors, and the coherence of the study groups. After setting up the model we conducted matching runs once a week from early May until the end of the academic year, that is mid-June in Hungary.

The first period of this application was very short, and we had an unexpectedly high number of volunteer mentors, but also an unexpectedly low number of registered students. Therefore the allocation problem was rather straightforward. Nevertheless, we used the data from this early period to generate instances that are realistic, and we present the simulations conducted on that data.

The pandemic situation became critical again in Hungary by October due to the second wave of infections, so on 6th of November 2020 all the secondary schools and universities were closed again, and thus we opened our allocation service again to link volunteering mentors with students.

The pandemic situation has been similar around the world, and especially critical since April in the USA. An early report on the American schools' responses to COVID-19 can be found in Harris et al. (2020).

In this paper we describe the optimisation aspects of our aforementioned joint NGO project of allocating voluntary mentors to students together with computational simulations on realistic instances. We believe that this paper provides an interesting case study with an advanced OR solution that could be used everywhere in the world to help allocating the volunteers to people in need in an efficient and fair way.

1.1. Related literature

Matching problems under preferences in two-sided matching markets have been widely studied in mathematics, computer science and economics, see e.g. a recent book on the algorithmic aspects of this topic by Manlove (2013). Beside the theoretical studies, practical applications have been designed and implemented in many areas, see a recent survey on this (Biró, 2017).

When both sides to be matched have preferences, then the concept of stable matchings was proposed in the seminal paper of Gale & Shapley (1962) and has been used since in many applications, such as resident allocation, college admission and school choice. However, there can be some special features that can make the stable matching problem computationally hard to solve. In this case one robust approach is (mixed) integer linear programming, that has been used recently for the hospital–resident problem with couples (Biró et al., 2014), ties (Delorme et al., 2019; Kwanashie & Manlove, 2014) and multiple objectives (Shimada et al., 2020), college admissions with lower and common quotas (Ágoston et al., 2016), and stable project allocation under distributional constraints (Ágoston et al., 2018). In this paper we also use MILP technique for solving the underlying optimisation problem.

There are also many application, where preferences of one or both sides do matter, but the solution is not necessarily stable or fair but optimal rather in some sense. Examples are the allocation of papers to reviewers (Garg et al., 2010), course allocation (Budish et al., 2017), or arranged marriages (Cao et al., 2010). Scheduling problems are also closely related, see a paper linking the two lines of research (Biró & McDermid, 2014).

Allocating mentors to students can be of dynamic nature so the literature on online matching is also related. Natural applications for online matching with preferences are deceased organ allocation (Agarwal et al., 2021; Mattei et al., 2018), allocation in social housing (Bloch et al., 2020; Leshno, 2019), electric vehicle charging (Gerding et al., 2019), or lending decisions (So et al., 2016).

There are also applications, which are dynamic in nature, but instead of online matching protocols, batch allocations are also used. An important example is refugee allocation, where preferences of one or both sides may be taken into account, together with some objective goals of maximising the likelihood of successful settlement of the refugee families, see Andersson et al. (2018); Bansak et al. (2018); Trapp et al. (2018). Similar approaches are used in the allocation of foodbanks (Prendergast, 2016).

An important example for an application where optimisation is used for the allocation are the kidney exchange programmes (KEPs), where kidney patients with incompatible donors may exchange their willing donor among themselves. Seminal work on IP models for KEP's is presented in Abraham et al. (2007) and Roth et al. (2007), a recent survey is Ashlagi & Roth (2020), and the European optimisation practices are summarised in Biró et al. (2021). It is interesting to note that online matching is used in the US, partly because of the competition in between multiple national programmes (Agarwal et al., 2019). However, in Europe the national programmes use batch allocations, by conducting the matching runs in 3–4 months regular intervals (Biró et al., 2019).

Finally, we note, that even though many of the features of our application are present in other matching applications listed above and there are also recent examples of the usage of MILP techniques in the field of matching under preferences, but we are not aware of any similar application, where mentors are allocated to students via optimisation techniques. Furthermore, we believe that our model is rather complex, which includes many novel features, especially the combination of pairs and groups in the solution. This latter feature causes computational challenges as demonstrated by our simulation results. The use of preferences from both sides without requiring the classical stability constraint is also unique in this application. The flexible timing of the matching runs is also interesting, since in most of the above mentioned applications either one single match run is done (e.g, in school choice and university admission) or the allocation is completely online (e.g., allocation of deceased organs). Therefore, the possibility of conducting batch matching runs is also an important special feature that we study in detail in this paper.

1.2. Our contributions

We describe the allocation mechanism that we designed with our partners and implemented in the application. The design is complex, it takes into account the preferences of both sides, and also several objective factors. The main novelty and challenge in our solution concept is that besides mentor-student pairs we also seek to form study groups, that makes the underlying optimisation problem more elaborate.

Our main theoretical contribution is an IP formulation that accommodates all the complex constraints and objectives of our model.

These results are complemented by computational experiments, where the generation of the instances is based on the real data that we collected in the first period of the application. In the simulation we analyse the effect of some optimisation policy decisions with regard to various performance measures. We also test the effects of having shorter or longer matching periods, and the possibility of giving priority based on the waiting time. The results of the simulation have been used to refine the design and optimisation policy of the application.

2. Preliminaries

First we give a general description of the model, then specify the variables and the input data for the preferences, priorities and further objective factors.

2.1. General description

We have a set of students $A = \{a_1, a_2, \dots, a_k\}$, mentors $B = \{b_1, b_2, \dots, b_l\}$, and subjects $S = \{s_1, s_2, \dots, s_m\}$, the latter being specified with the year as well (e.g., year 10 - Chemistry). We would like to form pairs (each consisting of one student and one mentor) and groups (each consisting of a set of students and one mentor) so that the online mentoring is conducted with these pairs and groups in weekly periods. For every pair and group formed we also specify the subjects that they are going to study and the amount of time that they are supposed to spend with each subject during a week.

As the input of the problem the students give the list of subjects that they need mentoring for in a preference order, specifying also the amount of time they wish to spend with a mentor per week (integer number between 1 and 3) for each subject. So a student may ask 2 hours of mentoring in Maths and 1 hour in Physics with higher preference for the former. The students may indicate why they need mentoring (e.g., being a child of a single mother who works in a hospital as a nurse), and about their objective circumstances with regard to their social background, such as the number of children/parent in the household, etc., that affect the priorities. Furthermore, they also provide information on the class they attend and whether they have ongoing online education for a particular subject, whether they are weak, medium or good students (by giving their final grade in the last semester), and whether they are willing to accept mentoring in groups or only in pairs. Finally, we ask what equipment they have at home (PC, tablet, smartphone).

Regarding the mentors, we ask in which subjects they are willing to mentor students (including the years), and whether they have preferences over the subjects, and some characteristic of the students, such as their age, how strong the students are in the subject, and their social status (some mentors can be especially keen to teach socially disadvantaged or poor-performing students, some might not be so). We ask the total number of hours that they wish to spend with mentoring per week and whether they are willing to do mentoring in groups.

The organisers of this project decided that some of the above mentioned priorities will only apply for such mentors who express their agreement. For instance, every mentor can tell whether she wishes to teach seriously disadvantaged students, and if they say no then no priority is given for those students when considering to allocate them to this mentor. However, if the mentor states, however, that she would be happy to teach seriously disadvantaged students then high weight is assigned. If the mentor is ignorant about this aspect, then a small weight is assigned for this priority factor. The same applies for the criterion whether the student is weak with low grades from the last semester. If a mentor expresses that she does not want to teach weak students then no priority is given for these pairs, but if she wishes to teach weak students then extra weight is given. Thus we take the preferences of the mentors into account for some of these controversial priorities, but for some others, e.g., number of children/parents in the family, we always give an extra priority. In this way the resulting pairs and groups are more likely to be mutually satisfying.

2.2. Notations: basic elements of the solution

The basic building block of the optimisation model is the set of possible paired mentoring activities E, where each *activity* e consists of a triple $e = (a_i, b_j, s_k)$. Activity e is possible if student a_i requested subject s_k and mentor b_j also offered s_k.

Let $S(a_i)$ and $S(b_j)$ denote the set of requested subjects by student a_i and offered subjects by mentor b_j, respectively. Furthermore, student a_i requested $q(a_i, s_k)$ hours per week in subject s_k and mentor b_j offered $Q(b_j)$ hours per week in total. We write that $(a_i, s_k) \in e$ if there is $b_j \in B$ such that $e = (a_i, b_j, s_k)$, and similarly, we write that $(b_j, s_k) \in e$ if there is $a_i \in A$ such that $e = (a_i, b_j, s_k)$.

In the description of a solution, let x_e denote the amount of hours scheduled for activity e in a pair. Furthermore, let y_e be a binary variable denoting whether activity e is performed in a pair, i.e. $y_e = 1 \iff x_e > 0$. In our practical application we restrict $x_e \in \{0, 1, 2, 3\}$, hence this is an integer variable in our MILP model.[1] Let P denote the set of pairs formed, i.e., $P = \{e \in E : y_e = 1\}$.

Besides pairs, we also allow the formation of groups, whose final set in the solution will be denoted by G. Each *group* $g \in G$ consists of a triple $g = (A^g, b_j, s_k)$, where $A^g \subset A$ is the subset of students assigned to group g with mentor b_j and subject s_k. In the application we restricted the amount of time for mentoring in each operating group to either 2 or 3 per week.

In our model and application, for simplicity, we assume that every mentor can have at most five groups to supervise in a subject. Thus, for every mentor b_j who is willing to teach groups in subject s_k, we create five *potential groups* $g_j^{k,1}, g_j^{k,2}, \dots, g_j^{k,5}$ with capacity c_j^k each. We introduce a binary variable $y_j^{k,t}$ to denote whether the potential group $g_j^{k,t}$ is realised, in which case there are at least two and at most c_j^k students assigned. Let $x_j^{k,t}$ denote the number of hours allocated for the weekly operation, where $x_j^{k,t} \in \{0, 2, 3\}$ (since if a mentoring group is operating then either 2 or 3 hours can be scheduled per week). Note that $y_j^{k,t}$ and $x_j^{k,t}$ are the variables of our mixed integer programming formulation.

For every activity $e = (a_i, b_j, s_k)$, we define a binary indicator variable y_e^t denoting whether this mentoring activity is performed in potential group $g_j^{k,t}$. Therefore, the set of students involved in this group will be $A^g = \{a_i \in A : e = (a_i, b_j, s_k), y_e^t = 1\}$. The following formula summarises the feasibility condition for realising potential group $g_j^{k,t}$ for every $t \in \{1, \dots, 5\}$.

$$y_j^{k,t} = 1 \iff 2 \le \sum_{e:(b_j,s_k)\in e} y_e^t \le c_j^k$$

For example, a potential group $g_j^{k,1}$ for mentor b_j can be on 7-year Maths (s_k) for at most 5 students ($c_j^k = 5$). If this group is realised ($y_j^{k,1} = 1$), a mentoring group $g = (A^g, b_j, s_k)$ will be formed where $2 \le |A^g| \le 5$ with $x_j^{k,1}$ hours per week, where $x_j^{k,1} \in \{2, 3\}$.

Regarding the personal mentoring hours in a group, as students in a group may have different requests, we estimate the actual mentoring hours by variable x_e^t, where we assume that $x_e^t \le q_j^{k,t}$ and $x_e^t \le q(a_i, s_k)$, where $e = (a_i, b_j, s_k)$.

2.3. Preferences, priorities and further objective factors

Here we describe the input more formally listing the information provided by the users with regard to their attributes and preferences.

[1] One can consider to set these variables to be continuous in the interval [0,3]. However, because of the study groups, we might get fractional values in an optimal solution, which is not allowed in practice.

2.4. Students' attributes and preferences

For every student a_i, we collect the following attributes and preferences:

- $year_i \in \{1, 2, \dots 12\}$: which year of study she attends.
- $class_i$: exact class she attends in the school (text).
- PA_i: preference list consisting of the subjects she requested mentoring (e.g., Maths, Physics, History). Let $rank_i(s_k)$ denote the rank of s_k in PA_i for $s_k \in S(a_i)$.
- RA_i: number of requested hours for each requested subject in the order of preferences. Note that in our LP model $q(a_i, s_k)$ denotes this constant for each subject $s_k \in S(a_i)$.
- $grades_i$: her grades from the last semester in the subjects requested, let $gr(a_i, s_k)$ denote the grade of a_i in subject s_k, that is a value between 1 and 5 in Hungary (5 for Best and 1 for Fail).
- $group_i \in \{0, 1\}$: the value is 1 if a_i is willing to accept mentoring in groups.
- $equipment_i \in \{0, 1\}$: the value is 0 if she has a smartphone or tablet and the value is 1 is she (also) has a laptop.
- $help_i \in \{0, 1, 2\}$: self-reported neediness, whether a_i has help at home (0= yes, 1= limited, 2= no)

When forming the groups, the similarities between the students can be important, therefore for those students willing to get mentoring in groups we define the following values for each pair of students a_i and $a_{i'}$.

- $sc_{i,i'}$: the value is 1 if they attend the very same class and 0 otherwise.
- $dr^k_{i,i'} = |q(a_i, s_k) - q(a_{i'}, s_k)|$: the difference between the requested amount of time in subject s_k.
- $dg^k_{i,i'} = |gr(a_i, s_k) - gr(a_{i'}, s_k)|$: the difference between their last year's grades in subject s_k.
- $de_{i,i'} = |equipment_i - equipment_{i'}|$: this value is 0 if they have the same equipment and 1 if they have different ones.

Furthermore, for those students wishing to get priority, we also ask question about her social background and circumstances, based on which we assigned the following scores to student a_i.

- $SD_i \in \{0, 1, 2, 3\}$: a measure showing how socially disadvantaged the student is (3 being the maximum value).
- NH_i: an index for not enough help at home with a value between 0.5 and 2.5 showing the number of children in the household per parent. Thus its value is a half-integer between 0.5 and 2.5.
- $WS_i \in \{0, 1, 2, 3\}$: showing how weak the student is based on her grades and repeated years.
- $CY_i \in \{0, 1, 2\}$: how critical the year is for the student, i.e, for last year of studies = 2, and for the penultimate year = 1.

2.5. Mentors' preferences

Here we describe what preferences the mentors are allowed to give on the subjects and students.

- PM_j: preference list of b_j on the subjects she offers for mentoring (e.g., Maths, Physics, History). Let $rank_j(s_k)$ denote the rank of s_k in PM_j for $s_k \in S(b_j)$. Note that this may be set differently for the three different age-categories, years 1–4, 5–8, 9–12.
- $YM_j \in \{0, 1, 2\}$: most preferred age of the student for b_j (0 = Year 1–4, 1 = Year 5–8, 2 = Year 9–12).
- $DM_j \in \{0, 1, 3\}$: whether b_j is willing to mentor socially disadvantaged students (0= rather not, 1= does not matter, 3= would be very keen).
- GPM_j: grade-preference, $= N$ if no preference is given, $= W$ for weak, $= M$ for medium, and $= S$ for strong students. From this information, we create the following constants: $PM_j = 0$ if $GPM_j = N$ and $PM_j = 1$ if $GPM_j \neq N$; $SM_j = 0$ if $GPM_J \in \{M, S\}$, $SM_j = 1$ if $GPM_J = N$ and $SM_j = 3$ if $GPM_J = W$; finally $WM_j = 0$ if $GPM_J = N$, $WM_j = 1.5$ if $GPM_J = W$, $WM_j = 3$ if $GPM_J = M$, and $WM_j = 4.5$ if $GPM_J = S$. Here, PM_j is an indicator whether b_j has grade-preferences; SM_j shows how willing b_j is to mentor a weak student; finally, WM_j is the best average grade of the student according to b_j's preference.

3. Optimisation with MILP technique

In this section we show how we can formulate our problem as a mixed integer linear program.

First we describe the basic constraints for the feasibility of a solution. We summarise the feasibility requirements as follows.

Feasibility constraints:

1. Use only mutually acceptable paired mentoring activities in the solution (i.e., the subject should be requested by the student and offered by the assigned mentor).
2. Only those mentors can have groups and only those students can be assigned to groups who expressed their willingness to teach or study in groups, respectively.
3. Have at least two and at most a limited number (c^k_j) of students in each group.
4. Weekly capacities of the mentors must not be exceeded.
5. The mentoring hours of a pair in a subject should not exceed the amount requested by the student, and never be more than 3. The mentoring hours per week for an active group is either 2 or 3.
6. Every student can be mentored by at most one mentor in each subject that she requested.

The first two conditions are automatically satisfied, since we only work with mentoring activities where the subjects are mutually acceptable by both parties (i.e., requested by students and offered by mentors). Similarly, we only create potential groups for those mentors who are willing to teach in groups and we only have variables y^t_e and x^t_e for those students who are willing to study in a group.

Regarding the pairs, the connection between x_e and y_e can be established with the following formula (1).

$$y_e \le x_e \le 3 \cdot y_e \text{ for every } e \in E \tag{1}$$

A similar formula (2) is added for each potential group, but with the minimum number of hours being 2.

$$2 \cdot y^{k,t}_j \le x^{k,t}_j \le 3 \cdot y^{k,t}_j \text{ for every } b_j \in B, s_k \in S(b_j), t \in \{1, \dots, 5\} \tag{2}$$

For the realisation of potential groups, we set the following conditions (3).

$$2 \cdot y^{k,t}_j \le \sum_{e:(b_j, s_k) \in e} y^t_e \le c^k_j \cdot y^{k,t}_j \text{ for every } b_j \in B, s_k \in S(b_j), t \in \{1, \dots, 5\} \tag{3}$$

Every activity can be performed either in a pair or in a group, described in constraints (4) below:

$$y_e + \sum_t y^t_e \le 1 \text{ for every } e \in E \tag{4}$$

The above constraint (4) is also enforced by the more general requirement (5) that every student can have at most one mentor in each subject:

$$\sum_{e:(a_i, s_k) \in e} \left(y_e + \sum_t y^t_e \right) \le 1 \text{ for every } s_k \in S(a_i), a_i \in A \tag{5}$$

For later use in the objective function, we introduce some new binary variables $\beta_i^k \in \{0,1\}$ indicating whether student a_i is involved in mentoring activity in subject s_k and $\gamma_i \in \{0,1\}$ indicating whether student a_i is involved in any mentoring activity. We link these variables with the basic variables as follows:

$$\beta_i^k = \sum_{e:(a_i,s_k)\in e} \left(y_e + \sum_t y_e^t \right) \text{ for every } s_k \in S(a_i), a_i \in A,$$

and let

$$\gamma_i \le \sum_{s_k\in S(a_i)} \beta_i^k \le M \cdot \gamma_i \text{ for every } a_i \in A,$$

where M is a large enough number, e.g., the number of subjects any student can possibly request mentoring (5 in our case). Note that the above binary restrictions of β_i^k make constraints (5) satisfied automatically, thus we can leave out these constraints from the model, as well as constraints (4).

The weekly capacity of the mentors should not be exceeded, as expressed with the following constraints (6).

$$\sum_{s_k\in S(b_j)} \left(\sum_{e:(b_j,s_k)\in e} x_e + \sum_{t\in\{1,\dots,5\}} x_j^{k,t} \right) \le Q(b_j) \text{ for every } b_j \in B \quad (6)$$

The mentoring hours of a pair in a subject should not exceed the requested amount by the student, that can be enforced with constraints (7) below.

$$x_e \le q(a_i, s_k) \text{ for every } e \in E, (a_i, s_k) \in e \quad (7)$$

When computing the volume of a solution, for groups we assume that the actual mentoring hours for a student (x_e^t) is bounded from above by her original request in this subject, and by the number of mentoring hours of the group. This can be formalised with the following two sets of constraints (8) and (8).

$$x_e^t \le q(a_i, s_k) \text{ for every } e \in E, (a_i, s_k) \in e, t \in \{1,\dots,5\} \quad (8)$$

$$x_e^t \le x_j^{k,t} \text{ for every } e \in E, e = (a_i, b_j, s_k), t \in \{1,\dots,5\} \quad (9)$$

Now, we turn to the objectives. Below we list all the possible objectives considered by the decision makers.[2]. We explain the rational behind the criteria and we formulate the corresponding linear terms for the objective function of our MILP model. The relative weights of these terms in the final objective function were set by the organisers of this application. (Note that the relative weights sum up to 100, so they can be interpreted as percentages showing the importance of the objective criteria.)

As the most important parameter, let w^g denote the discount for a mentoring activity being realised in a group, as opposed to a pair. In our default setting, we use $w^g = 0.7$, which means that each mentoring hour in a group counts 0.7 hour in a pair. This parameter highly affects the share of groups in the final solution, as we will also demonstrate later in the simulations.

1. **Number of students allocated.** Allocating mentors to as many students as possible.
   Rational pros: We would like to involve as many students as possible in the mentoring.
   Rational cons: If this objective is dominating then we would have many mentoring activities with 1 hour only that can be inefficient for both students and mentors.
   Relative importance in the application: 0 (i.e., this criterion was decided not to be considered as objective, only monitored)
   Linear term: With the use of variables γ_i, we can simply express this objective as follows.
   $$\sum_{a_i\in A} \gamma_i$$
2. **Number of pairs and groups created.** Maximising the number of pairs and groups realised with different weights for pairs and groups.
   Rational pros: We would like to create as many pairs and groups as possible, to create the links between the parties.
   Rational cons: If this objective is dominating then we would have many mentoring activities with 1 hour only for pairs and 2 hours for groups that can be inefficient for both students and mentors.
   Relative importance in the application: 0 (i.e., this criterion was decided not to be considered as objective, only monitored)
   Linear term: Similar to the formula for β_i, we express this objective as follows.
   $$\sum_{e\in E} \left(y_e + \sum_t w^g \cdot y_e^t \right)$$
3. **Volume.** Maximising the number of mentoring hours realised with different weights for pairs and groups.
   Rational pros: The number of mentoring hours is the most important measure.
   Rational cons: The solution might be unbalanced, some students may get many hours while some others may get none.
   Relative importance in the application: 50
   Linear term: Similarly to the previous formula we now use the hours rather than the indicator variables which can be considered the volume of the solution.
   $$\sum_{e\in E} \left(x_e + \sum_t w^g \cdot x_e^t \right)$$
   As the volume will be the main objective in our optimisation, we will set the relative weights for each mentoring activity accordingly. Let w_e denote the final weight of activity e in a pair and w_e^g denote the final weight of an activity in a group. We suppose that $w_e^g = w_e \cdot w^g = w_e \cdot 0.7$ in our case. The final weight w_e will be a sum of weights with respect to different objectives. The first objective is the volume, that we weighted 50 for all activities, so let $w_e^w = 50$ for every activity e.
4. **Preferences.** Satisfying the preferences of the students with regard to the subjects, and the preferences of the mentors on the subjects and on the age of students.
   Rationale: The higher the need of the student for a subject is the more important is for her to get help, and the preference of the mentor also is to be taken into account for the subject and so is the age of the student supervised.
   Relative importance in the application: 10
   Linear term: Each activity $e = (a_i, b_j, s_k)$ will get an additional weight according to the preferences of the students and mentors that we denote by w_e^p. Let $w_e^p = (6 - rank_i(s_k)) + (6 - rank_j(s_k)) + 3 * agepref_i^j$, where each of the first two terms gives a value between 1 to 5, depending on how preferable this subject is for the student/mentor, and the last term gives 3 if the age of the student is preferred by the mentor (among years 1–4, 5–8, or 9–13).
5. **Group cohesion.** When forming groups, we shall preferably have students from the same class, with the same type of equipment, have their former grades as close as possible, and have the number of scheduled hours to be close to what the assigned students requested in that subject.
   Rationale: Two students from the very same class are preferred to be put into the same mentoring group, as they receive

[2] We note that additionally one might wish to consider the genders of the mentor-student pairs, or whether they live in the same city, geographic area, or maybe if they attend the same school in the case of student mentors.

the same distance education from their home school. Forming groups for students with similar strength can improve the efficiency of mentoring. Finally, the requested hours by the students in a group should be close to the scheduled hours. Regarding the equipment, if a student in a group has a laptop and another only a tablet or smartphone then the possible interactions can be limited between them and the mentor (we consider tablets and smartphones to be equally useful).
Relative importance in the application: 15
Linear term: For any two students a_i and $a_{i'}$ who are both willing to accept mentoring let us introduce a binary variable $z^p_{i,i'}$ for every potential group $p = (b_j, s_k, t)$, where both a_i and $a_{i'}$ could belong to, i.e., if there exist $e = (a_i, b_j, s_k)$ and $e' = (a_{i'}, b_j, s_k)$ The indicator variable $z^p_{i,i'} = 1$ if both a_i and $a_{i'}$ are assigned to p in the solution. This can be achieved with the following new constraints.

$$z^p_{i,i'} \geq y^t_e + y^t_{e'} - 1 \text{ for every } p = (b_j, s_k, t), e = (a_i, b_j, s_k), e' = (a_{i'}, b_j, s_k) \tag{10}$$

$$z^p_{i,i'} \leq y^t_e \text{ for every } p = (b_j, s_k, t), e = (a_i, b_j, s_k) \tag{11}$$

$$z^p_{i,i'} \leq y^t_{e'} \text{ for every } p = (b_j, s_k, t), e' = (a_{i'}, b_j, s_k) \tag{12}$$

Furthermore let $z^k_{i,i'} = \sum_{p:s_k \in p} z^p_{i,i'}$, where $z^k_{i,i'}$ is the indicator variable showing whether a_i and $a_{i'}$ are in the same group for subject s_k.
The accumulated weight of the group coherence criteria is

$$GC(z) = \sum_{a_i, a_{i'} \in A} \sum_{s_k \in S} (10 \cdot sc_{i,i'} - 2 \cdot de_{i,i'} - dg^k_{i,i'} - dr^k_{i,i'}) \cdot z^k_{i,i'}$$

6. **Students' priorities.** Giving priority to certain students in need (depending on the preferences of the mentors for some criteria, such as social status and student past performance).
Rationale: Social welfare can improve if the students in need get mentored. However, the criteria of being socially disadvantaged or being a weak student can be controversial for some mentors, so we allow them to express their willingness to get paired with such students.
Relative importance in the application: 20
Implementation with weights:
The combined social priority weight of $e = (a_i, b_j, s_k)$, denoted by w^s_e, is as follows:

$$w^s_e = 3 \cdot SD_i \cdot DM_j + WS_i \cdot SM_j + (1.5 - |gr(a_i, s_k) - WM_j|) \cdot PM_j + 2 \cdot CY_i + (NH_i + help_i)$$

7. **Multiple subjects for a pair.** When the same mentor-student pair is involved in paired mentoring activities in multiple subjects then this assignment is preferable to the case where this does not happen.
Rationale: It should be avoided that too many different mentors supervise the same student.
Relative importance in the application: 5
Linear term: We introduce a new binary variable m^j_i to denote whether a_i is mentored by b_j in any subject in a pair with the following constraints.

$$m^j_i \leq \sum_{e:(a_i, b_j) \in e} y_e \leq 5 \cdot m^j_i \tag{13}$$

We shall minimise the number of mentoring pairs with a weight w^m, that is let

$$MP(y) = w^m \sum_{i,j} m^j_i$$

where we set $w^m = 5$ in our application.

3.1. Final objective function

For $w_e = w^w_e + w^p_e + w^s_e$ (and $w^g_e = w_e \cdot w^g$), the final objective function is:

$$\sum_{e \in E} w_e \cdot x_e + \sum_t w^g_e \cdot x^t_e + GC(z) - MP(y) \tag{14}$$

To summarise, the main objective is to maximise the volume of the mentoring activities while considering the preferences and the social priorities of the students, improving group cohesion, and decreasing the number of mentors per students.

4. Simulations: Data generation

During the first operating time of the allocation scheme (1 May to 15 June 2020), the number of students who registered on the webpage was 14, while the number of mentors was 56. Because of the low number of students' registration, we could not judge the true potential of the model and application. Therefore we decided to conduct computational experiments on a mix of real and generated data.

Because of the low amount of observations, the main goal of our simulation was to test our MILP model, rather than to give accurate prediction based on real data. We still tried, however, to use the data available to make the generated data as realistic as possible.

We computed the correlations amongst the mentor variables. Having found only weak correlations, we generated the parameters by rolling a biased die independently for each parameter and observation (student and mentor). We used the available data to estimate these biases. In this section, we describe the way we generated each parameter.

4.1. Students

- **School-ratio:** 14 students registered from 9 different schools. Therefore in the simulator, we considered the student-school ratio to be at most 67%. So when we considered 100 students, we generated 67 schools, and for each student, we selected a school from the whole set of schools with replacement.
- **Number of subjects:** The average number of subjects requested per student was 2, with 4 as maximum. We used the following distribution for the amount of subjects: (1: 50%; 2: 30%; 3: 10%; 4: 10%).
- **Time ($q(a_i; s_k)$):** The average time required by the students for mentoring per week in a subject was around 2 hours with a minimum of 1 and maximum of 4. We used the distribution (1: 33%; 2: 33%; 3: 25%; 4: 9%).
- **Grades:** For the distribution of grades, we got almost every possibility from the data, except grade 1, which means Failed in Hungary. In the simulation, we generated all types of grades uniformly with the addition of 0, where 0 means no response (which was 25% of the real cases).
- **Group:** 9 out of 14 students selected the possibility of getting mentoring in groups. Hence we set $\frac{2}{3}$ for the probability of a student accepting group-mentoring.
- **Help:** For the three possible values we received 5-5-4 responses, respectively. Therefore, we decided to generate these values randomly with equal probability.
- **Equipment:** All of the participants chose 0. We left this parameter out from the simulation because it is not crucial for the optimisation model.
- **Year:** Most of the registered students were from years 4–8 (92%), only one student was in year 11. Since the programme started at the end of the spring semester and after the matriculation exam, we assumed that the secondary school students

Table 1
Families by number of children, (http://www.ksh.hu/thm/2/indi2_1_4.html?lang=hu) .

Number of children	probability
1	69.2%
2	23.9%
3	5.2%
4 or more	1.7%

were underrepresented in the application. Therefore we generated the students' years uniformly from years 4 to 12.

- **Prior1 (**SD_i**):** 13 students out of 14 chose option 0, and the last remaining student chose option 1. Therefore we generated a higher ratio of underprivileged students by using the (0: 65%; 1: 20%; 2: 10%; 3: 5%) distribution.
- **Prior2 (**NH_i**):** In the generator, we used the result of a 2016 census of the Hungarian Central Statistical Office (http://www.ksh.hu/thm/2/indi2_1_4.html?lang=hu) to generate the data. According to this census the percentage of single-parent families is 18.3%. For the number of children, we generated the data according to Table 1 (for *4 or more* we considered simply 4). Therefore, for each student, we generated the number of parents and the number of siblings independently, according to the above statistics.
- **Prior3 (**WS_i**):** In the real data, 64% of the students had 0 points, 28% had 2 and only one student had 3 points (no student got 1 point for this variable). We approximated the values of this parameter with the Poisson distribution, where we set the expected value equal to the mean of the real data (0.786).
- **Prior4 (**CY_i**):** This point depends directly on the year of the student, so it is computed accordingly.
- **Matriculation:** This depends on the year of the student. Therefore we only considered this variable when the student was from year 11 or 12. Most students officially take the matriculation exam in year 12 in Hungary, however, there is an option to advance some exams before the year of matriculation. Hence we randomised this value for the students from year 11, by assuming that 40% of them do not want to prepare for the matriculation exams (so they got the "N"-letter, meaning no matriculation), 40% going for the basis exam and 20% of them choosing the advanced exam.

4.2. Mentors

When we generated the characteristics of the mentors, we used the distributions taken from the real data. Since 56 mentors registered into the programme, we have not made as many assumptions as in the case of the students.

- **Group:** 54% of the mentors agreed to the possibility of mentoring in groups.
- **Time (**$Q(b_j)$**):** For the time capacity of the mentors, first we generated ranges with a distribution, and then we chose uniformly the exact value from the range selected. The time-range of a mentor was 1–3 hours with 40% probability, 4–6 hours with 40% probability, and 7–10 hours with 20% probability.
- **Social (**DM_j**):** We used the (0: 50%; 1: 40%; 3: 10%) distribution, in line with the real data.
- **Weak (**GPM_j**):** Here 85% of the mentors chose option *N*, therefore we also used 85% for generating option *N* and 5% for each the other three options.
- **Student-Age (**YM_j**):** We used the (0: 5%; 1: 20%; 2: 15%; N: 60% (no preference given)) distribution, based on the real data.

4.3. Subjects

Overall, there were 15 different subjects offered for selection. We generated the distribution of the requested subjects according to the distribution of the subjects offered by the mentors. The assumption behind this is that the demand and the supply of the subjects shall be balanced in the long run. However, we also added a random noise to modify the distribution for every instance.

Not all of the subjects are available for each year. Therefore the distributions are normalised for each year with regard to the subjects available. A student can request mentoring in multiple subjects. According to the data, the maximum number of subjects requested was 4 with an average of 2. As we already described, we used the ($1 = 50\%$, $2 = 30\%$, $3 = 10\%$, $4 = 10\%$) distribution to generate the number of subjects requested by each student, and then with the consideration of the student's year, we picked the subjects randomly with a distribution that is close to the distribution of the subjects offered by the mentors.

The mentors can belong to multiple classes. In the data, the average number of subjects per mentor was 2.9. However, the maximum number of the subjects was 9 for some mentors. Therefore some extreme values have increased the average a lot. We generated the number of subjects of each mentor with respect to their total time offered. If the mentor's total time was less than 4 hours then we generated 1–3 subjects uniformly. If the total time was less than or equal to 6, then we generated 1–4 subjects uniformly. Finally, in the case of time at least 7, we generated 1–5 subjects. Whenever we allocated a subject to a mentor, we assumed that the mentor is willing to teach students of all age in this subject.

4.4. Comparison of generated and real data

To compare the simulated data to the real data, we generated 1000 instances with the same amount of mentors (56) and students (14) as in the real data. Then we checked whether the values of the real data are within the interquartile range of the values of the generated data.

Regarding the evaluation measures, only the *Social*-points were not inside this range with any type of objective value. This is reasonable since in the real data, we have not received enough values (see, for example, Prior 1 and 2). Therefore we adjusted the distributions. Hence the generated data produced, on the average, higher results than in reality.

Also, when we considered the group-weight of 1 in the objective function, several other measures were also outside the interquartile range. The reason behind this instability is the few numbers of students in the real data.

5. Simulations: single run

In this section first we describe how we simplified the evaluation measures by clustering them based on one-shot simulations, and then we also present the performance analyses with regard to some basic parameters.

We used an AMD Ryzen 5 2600 Six-Core CPU 3,40 gigahertz computer with 16 GB DDR4 RAM for calculations. We ran the program in Python 3.7.6. and used the Gurobi 8.1.0 solver for the optimisation.

5.1. Generation of instances

For the basic setting we generated 100 large instances with 80 students and 40 mentors in each, and we conducted a single match run for each of them. This setup differs from the real data, since the student/mentor ratio is much higher in the generated data. Our

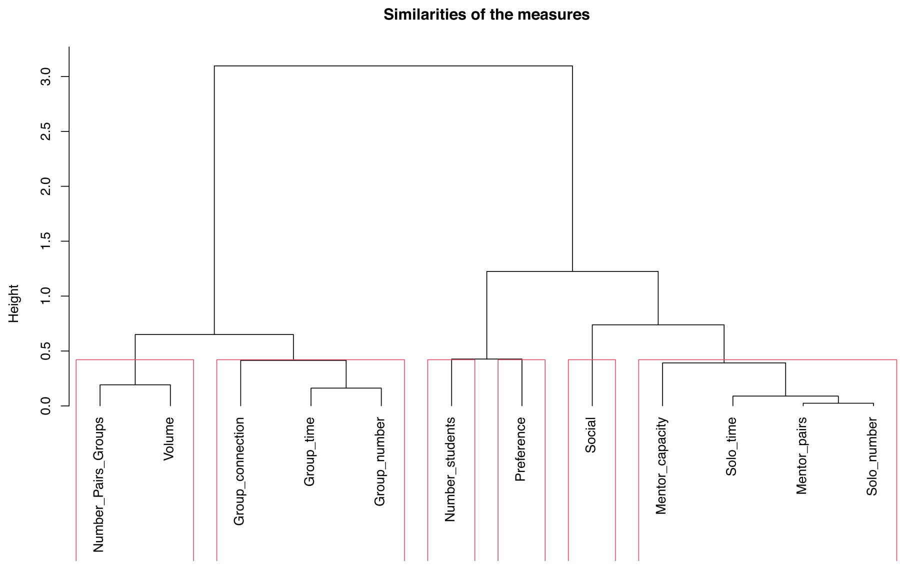

Fig. 1. The dendrogram of the measures. The measures are described in Section 3. `Number_of_students`: Objective 1 `Number_of_Pairs_and_Groups`: Objective 2; `Volume`: Objective 3; `Preference`: Objective 4; `Group_connection`: Objective 5; `Social`: Objective 6; `Mentor_pairs`: Objective 7; `Solo_time/Group_time`: Evaluation measure 1; `Solo_number/Group_number`: Evaluation measure 2; `Mentor_capacity`: Evaluation measure 3.

aim was to analyse the effect of using different objective functions when the programme has an ideal student/mentor ratio.

Then we adjusted the main parameters to create alternative solutions, as follows.

5.1.1. Alternative parameters

- we modify w^g from 0.7 to 0.5, 0.6, 0.8, 0.9, 1
- for $w^g = 0.7$ we modified the preference based weights from the default setting $w_e^p = (6 - rank_i(s_k)) + (6 - rank_j(s_k)) + 3 * agepref_i^j$ to b) $w_e^p = (6 - rank_i(s_k))^2 + (6 - rank_j(s_k)) + 3 * agepref_i^j$, and to c) $w_e^p = (6 - rank_i(s_k))^2 + (6 - rank_j(s_k))^2 + 3 * agepref_i^j$

Thus we generated five times 100 instances by varying w^g and another two times 100 instances by changing the preference based weights. So altogether we considered 800 instances. The solver always returned an optimal solution. The run time for solving the MILP model for these instances was relatively short, the average run time was 167 seconds with a maximum of 3 hours and 12 minutes.

5.2. Clustering the evaluation measures

Besides the seven objectives given in Section 3, we considered the following five performance measures in the evaluation of solutions.

5.2.1. Additional evaluation measures

8. number of pairs
9. number of groups
10. number of paired mentoring hours
11. number of group mentoring hours
12. total capacity of the mentors used

We analysed how the above described five evaluation measures behave together with the seven different objective functions, that we presented earlier. For the evaluation of the different objective functions, we decided to reduce the dimension of the twelve measures. So, we calculated the similarities between the various measures with hierarchical clustering. All of the results were considered in the estimation. Fig. 1 shows the dendrogram of the measures.

We decided to reduce the dimension of the 12 measures to 6 (as the red rectangles show in Fig. 1). We chose the six dimensions, because according to Principal component analysis, with 6 component 96% of the total information can be saved. Additional component only increased the saved information by less than 2%.

According to hierarchical clustering, we evaluated the `Number_students`, `Preference` and `Social` measures independently from the other measures.

From the `Mentor_capacity`, `Solo_time`, `Mentor_pairs` and `Solo_number` measures with using the Factor analysis method, we created a factor variable. We covered 84% of the total information in this factor. We named this factor as *solo-factor* because it is related to the measures of the non-group classes. For every measure a higher value increases the accumulated factor score. Interestingly the number of Mentor-student pairs (Objective 7) behaves similarly to the solo-class time and number.

We created another factor for the measures of `Group_connection`, `Group_time`, and the `Group_number`. We named this factor as *Group-factor*. We could cover 73% of the total information inside one factor. Higher `Group_connection` value decreases the Group-factor's score, while for the other two measures higher value will increase it.

Finally, from the `Number_of_Pairs_and_Groups` and `Volume` we created the so-called *Quantity-factor*. For all of these measures higher values increase the factor's score. We could cover 81% information within this factor.

5.3. Analyses of one-shot simulations

Fig. 2 presents the distributions of the three factors as well as the Number of students, Social and Preference scores of the different objective functions.

On the graphs of the Solo-factor and Group-factor, the influence of group-weight ($w^g = WG$) is clearly visible. As the group-weight increases, the Solo-score decreases and the Group-score increases. The two quadratic-preference models had very similar results in these cases compared to the original `WG_0.7` model, although there was a small decrease in the group-score.

The Quantity factor's score increases with the group-weight. Note that this is not surprising, since in both corresponding objectives (Number of pairs and groups, Volume) w^g is explicitly in-

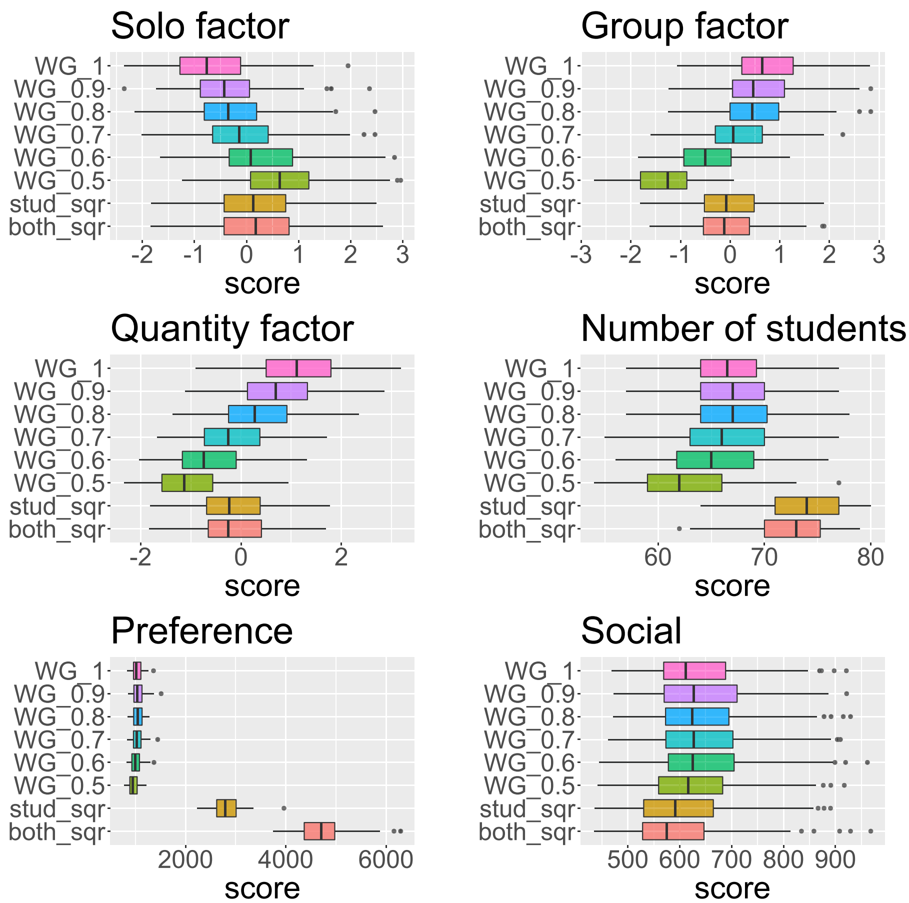

Fig. 2. The distribution of the 3 factors and the Number of students, Preference and the Social measures.

cluded in the formula. However, the simulation result shows, however, that this objective is not affected by the quadratic preferences.

Group-weight also increases a bit the number of matched students. However, the best solution for this measure is when the preference of the students was considered quadratically.

Naturally the Preference measure is very high for the two quadratic preference cases, since the objective functions are explicitly influenced in those cases. It seems, however, that the group-weight parameter did not have any effect on this measure.

Social points appear to be very stable, the group-weight does not have much effect on the social aspect of the programme. The quadratic preference cases, however, have a slightly worse result for this measure.

5.4. Sensitivity analysis of computational time

The complexity of solving the MILP comes from the possibility of study groups, since if only paired mentoring is allowed then the problem is reduced to the well-known assignment problem (Kuhn, 1955) that is solvable in polynomial time.

To investigate how the possibility of group mentoring changes the computational time, first we generated instances with different supply for group mentoring, by varying the ratio of the mentors willing to teach in groups. We considered 50 mentors and 100 students in each instance. First we generated ten instances with zero willingness of the mentors to teach groups, then in the following ten instances, we increased this percentage by five percent, and continued up to 100 percent.

Fig. 3 shows the average computational times of the models when changing the ratio of mentors willing to teach groups. The pale colour shows the range between the minimum and maximum computation times.

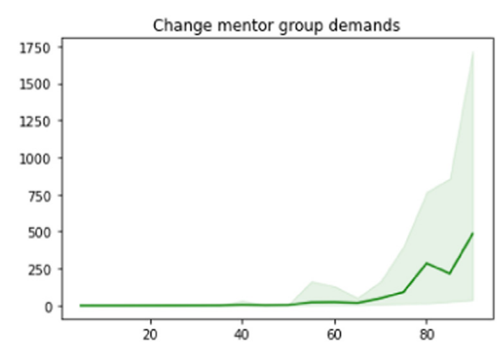

Fig. 3. Increasing the willingness of the mentors to teach groups.

It is noticeable that when most of the mentors would be willing to teach groups, then the running time is significantly longer than when none or few of the mentors would. When the ratio was less than 50%, the computational time remained low, and almost the same, but over 60% percent we see a significant increase.

The student-mentor ratio can also affect the running time of the model. If there are more mentors than students, then it is generally easier to create pairs. If the number of students is much higher than the number of mentors, then the optimal solution will be more oriented towards group education. In the second test we generated instances with exactly 100 participants, and changed the ratio of students and mentors from 5:95 up to 95:5. Fig. 4 presents the running times of the solutions. On the left side, the results for

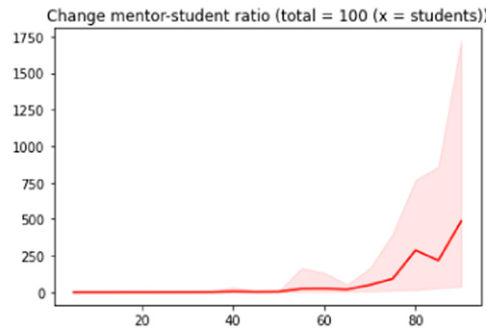

Fig. 4. Total of 100 participants, where axis x shows the number of students.

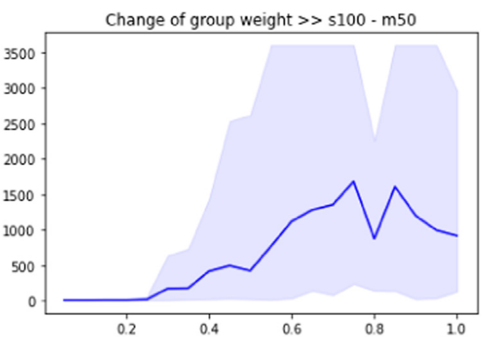

Fig. 5. Increasing the group weight, with 100 students and 50 mentors.

5 students and 95 mentors are shown, and on the right side the instances have 95 students and 5 mentors.

As the ratio of students increases, the computational time also increases, since more groups can be created when there are fewer mentors and more students.

Finally, in the third test, we considered how the group-weight w^g in the objective function affects the running time of the solution. Similarly to the previous case, we generated the instances with a fixed number of 100 students and 50 mentors. Then we calculated the optimal solutions with different group-weights in the range 0 to 1 for ten generated instances for each value.

As Fig. 5 depicts, the group-weight also influences the computational time. When the group weight was small, namely less than 0.25, then the running time was significantly smaller than for higher weights. Interestingly, the solution with weight 1 was not the slowest in general, but rather when the weight was slightly less than 1.

6. Simulations: Dynamic allocation

We investigated also how the frequency of the matching runs affects the results in a dynamic setting. We considered the first 300 days of the programme, by generating a registration date for each student and mentor. We assumed that both the students and mentors joined the application random uniformly distributed in the period considered. We also generated a leaving day for each applicant as follows. Presumably, if a student does not get any mentor within a reasonable time in this programme, then he or she may well seek mentors using other channels. We assumed that the stay of the students in this programme is normally distributed with expected value 14 (days) and standard deviation 2 (but they remain at least seven days). The mentors may also leave a programme if they do not get any student soon enough after registration, however, we assumed that they are be more patient than the students. In particular, they leave the programme 3 weeks after registration on the average (with a minimum of two weeks). Again we used normal distribution with setting two days for the standard deviation.

We also extended the stay of those students and mentors, who got matched. For any student, who was in a solution of a matching run, we added seven more days to their leaving time. We extended the staying time of the matched mentors by 14 days. Fig. 6 shows the flow of a matching run.

First, we set up the pool of students and mentors of a matching run. A pool of students consists of those students who have already registered into the programme, but have not left it yet and have demand for some subjects. The pool of the mentors are those who have time for teaching, and already registered but have not left the pool yet.

Then we calculate the optimal matching according to the policy setting and we remove the satisfied subject from the list of the demanded subjects of each selected student. For the allocated mentors, we decrease their total time offered for teaching with the allocated time.

Then we increase the remaining time of stay of the allocated students and mentors, with 7 and 14 days, respectively.

We considered four different frequencies for the matching-runs: 1, 2, 7 and 14 days. We evaluated 100 generated instances according to these four frequencies. In the optimisation model, we considered the Final objective function (14) with the default group-weight $w^g = 0.7$.

Since we had no access to the registration dates and duration of stays of the students and mentors, we generated two different set of instances with regard to the frequency of the arrivals. We considered instances where 1 and 4 were the average number of students joining the programme per day. Therefore, for an instance of type 1, we generated 300 students, and for an instance of type 4, we generated 1200 students for the period of 300 days. The students register into the programme uniformly at random, hence 1 and 4 students register daily on average, respectively.

In each case, we assumed that the mentors arrive half as frequently as the students. Hence in instances of type 1, one mentor arrives in every two days, in instance of type 4, two mentors are expected to join the programme every day on the average.

Fig. 7 presents the solo-groups formed in different examples.

In both cases, more frequent runs resulted in more solo classes. Between the 1 and 2 daily runs, the difference is not large, however as the matching runs become less frequent, the difference is visible. Fig. 8 presents the results of the number of groups formed.

In both cases, less frequent runs resulted in more groups. Therefore independently from the sizes of the matching problems, more frequent runs were better with respect to paired-mentoring, and less frequent runs were better for forming more groups.

How about the quality of the solutions of each frequency? In the previous figures, we focused on the number of pairs and groups. For the quality of matchings, first we considered the Social aspect of the programme. Fig. 9 presents the sum of the Social points in each type of matching runs.

The Social score did not depend on the frequency of the matching runs that much when in average one student registers per day. A small decrease is noticeable as runs become less frequent in this case.

However, when the average number of daily student registration is 4, then the trend changes. The reason behind the change of trend is the higher number of groups.

Fig. 10 presents the volumes of optimal solutions. With fewer students, forming a group is more complicated. Hence more frequent runs have better social points, because of the solo-groups. With a larger pool, more groups can be formed, therefore we see an increase in the Volume, as well as in the Social score.

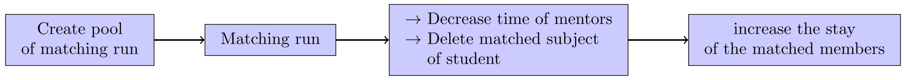

Fig. 6. Flow of a matching run.

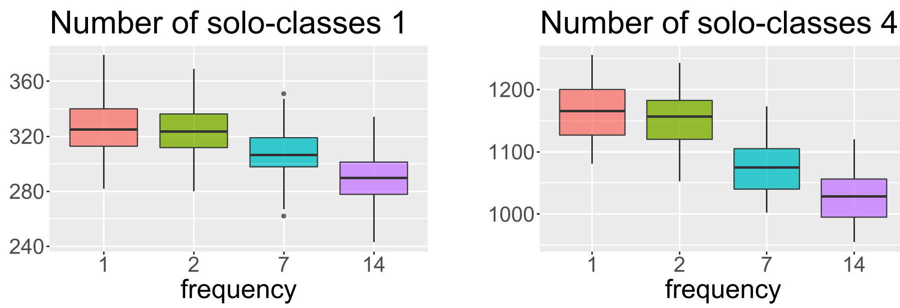

Fig. 7. The formed solo-groups.

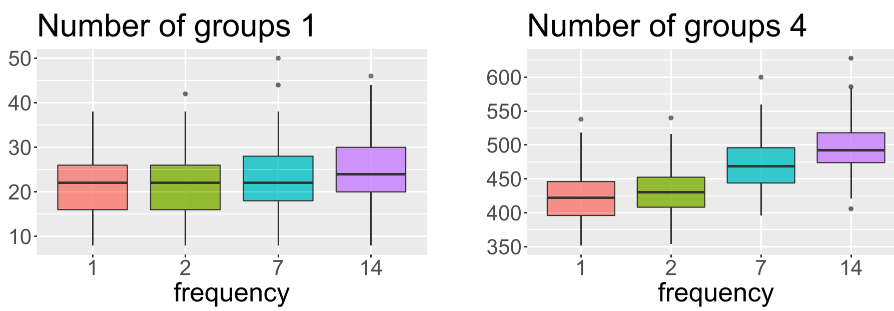

Fig. 8. The formed groups.

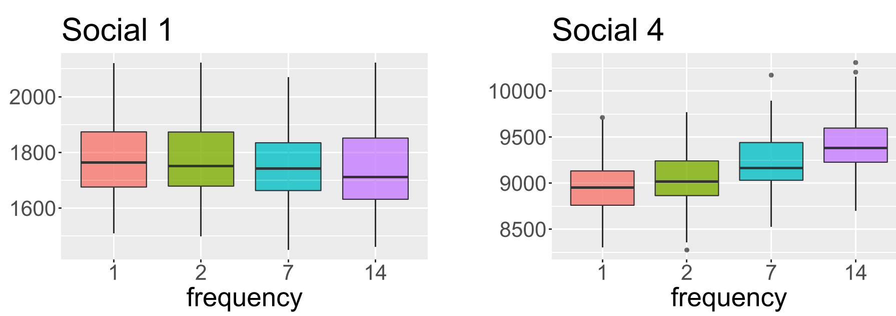

Fig. 9. The sum of social scores in each example.

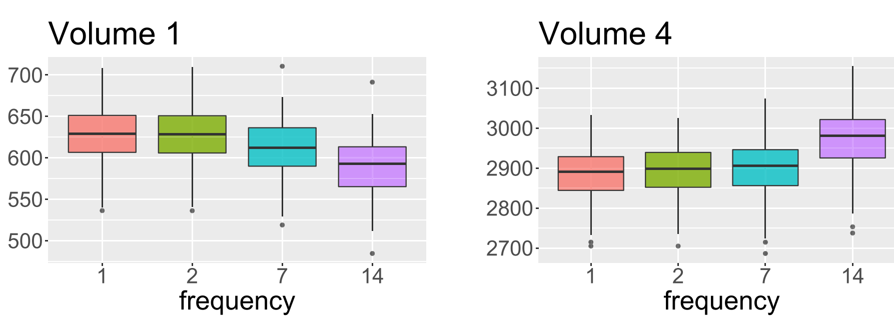

Fig. 10. Volume.

6.1. Consideration of the waiting-time

As in other dynamic allocation systems of bounded length of duration of stays (e.g., organ allocation) one may try to improve the solutions with a prioritisation based of *waiting times*.

In the following part we investigate how the results change if we prioritise those students and mentors, who registered earlier to the programme. It may decrease the social-scores and the preference-scores of the matchings, but there may be less early quits from the programme and also less unmatched participants.

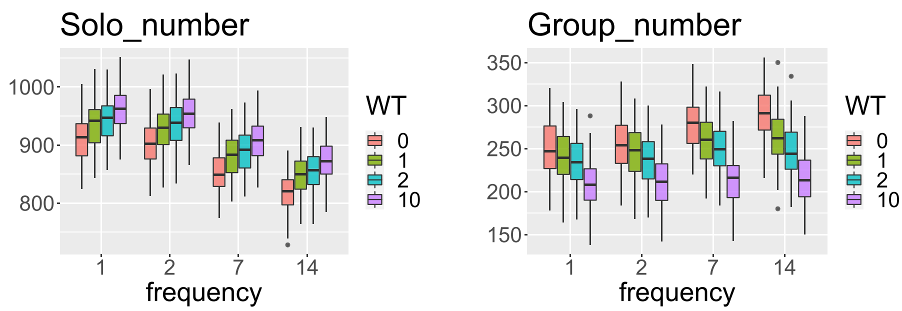

Fig. 11. The Solo and Group results with 0,1,2,10 weight on the waiting time.

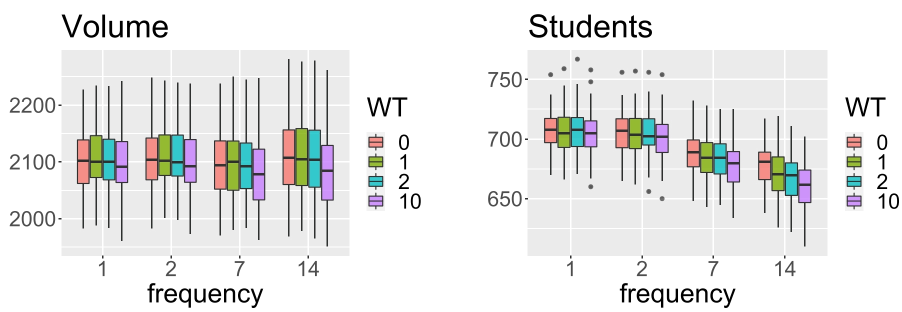

Fig. 12. The Volume and Number of students results with 0,1,2,10 weight on the waiting time.

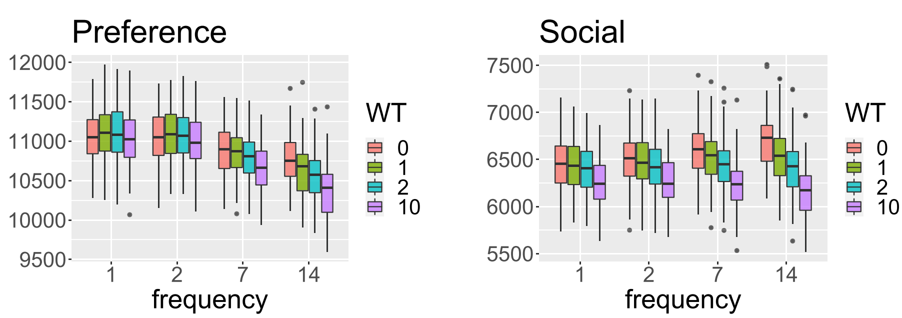

Fig. 13. The Preference and Social results with 0,1,2,10 weight on the waiting time.

Therefore we modified the weight of the activities, to $w_e = w_e^w + w_e^p + w_e^s + w_e^t$, where

$$w_e^t = WT\Big((t^r - t_i^0) + (t^r - t_j^0)\Big) \quad (a_i, b_j) \in e \tag{15}$$

Here t^r denotes the day of the matching run, t_i^0 is the registration date of student a_i and t_j^0 is the registration date of mentor b_j. We also use a weight for the days passed, that we denote by WT.

To test how the prioritisation by waiting time changes the solutions, we considered a 300 days period again with an average of three students registering into the programme in every day. We set 0,1,2 and 10 for the weight of the waiting time priority regarding all the match-frequencies that we investigated earlier.

Fig. 11 presents the Solo-groups' and groups' distributions of the results.

With the increase in the weight of the waiting time priority, the number of pairs increased, but the number of groups decreased. We can observe a similar trend for each match-frequency, hence giving priority for the waiting time is good for forming more pairs.

Overall, we can notice a small decrease in the Volume, as Fig. 12 presents. For WT = 1 and 2 this reduction is negligible, but for WT = 10 it is significant. Thus giving priority by the waiting time, we lose more mentoring time with the decreased number of groups than what we gain with more pairs.

The same figure depicts the change in the number of students as well. In every run-frequency, a higher WT resulted in a decrease in the number of students allocated.

Fig. 13 presents the Preference and Social scores. In both cases, higher WT resulted in a relapse. It is connected to the decrease in Volume since fewer mentoring hours in general means worse Preference and Social scores.

Hence the prioritisation by the waiting time increase only the number of pairs and decrease many other aspects of the programme. The setback in these values is caused by the multiple demands and offers. For example, a student may request two different subjects. If there is a match for her first subject then she remains in the programme with her second remaining subject with an extended duration of stay, and so in a later run she may well receive a mentor for her second subject as well due to her increased priority instead of allocating a newly registered student for her first subject. In fact, the newly registered student can even have higher Social score, or other scores can also be better for her, but the high waiting time priority for the aforementioned student overrules these scores. Thus the newly registered student may not even get a mentor, whilst the earlier matched student will get multiple mentors. Therefore there is a setback in both match quality and also in quantity to a certain extend.

To reduce the setback, caused by the multiple demands and offers by the members, we also investigated how the solution will

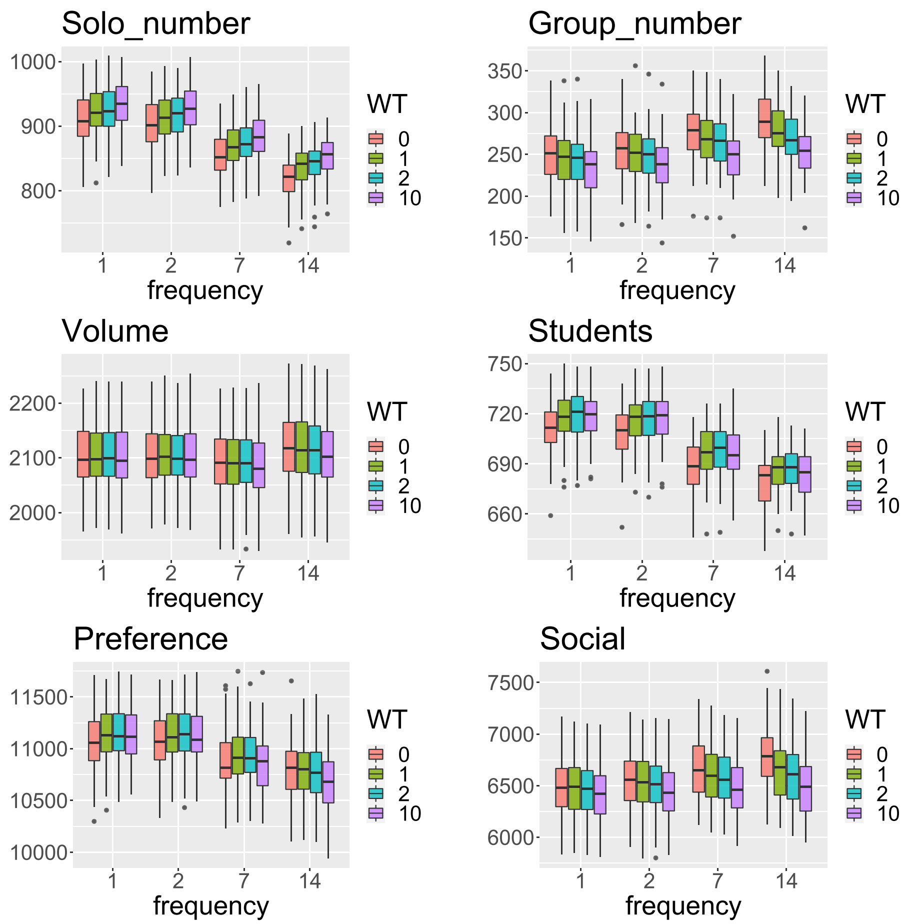

Fig. 14. The results with weights 0,1,2,10 for the waiting time regarding the first preferred subjects only.

change, if we only consider the extra priority on the waiting time for the first preferred subjects of the students. Fig. 14 presents the results of these models, with the same setups as earlier.

In the pairs and groups, the effect was similar to the earlier investigated case. The weight on the waiting time of the first subject also increased the number of pairs and decreased the number of groups. However, the change is noticeably smaller. Therefore the Volume appears to be more stable, but still has a slight reduction as the WT increases. However, the trend in the number of students changed. In this case, a higher weight on the waiting time of the first subject increased the number of students matched. Because each student has one subject with this weight, those students who have not been selected yet has a higher chance to get matched than before.

Regarding the preferences, we can notice a small increase when the runs occur every day or every two days. The score decreases, however, when the runs happen less frequently. The reason behind the change in the trend may be due to the reduction in the number of groups. In general, more frequent runs resulted in more pairs and fewer number of groups. The weights for the waiting time also increased the number of pairs and reduced the possibility to form a group. More pairs with higher preferences increased the preference score, but having fewer groups decreased it.

The Social score decreased for every run frequency as we increased the WT. However, the effect is much smaller compared to the case when every subject was weighted.

7. Conclusion

In this paper we have described the optimisation aspect of a joint NGO project for allocating voluntary mentors to students. By taking the participants' preferences into account we aimed at creating desirable pairs and study groups by using integer programming techniques for solving the dynamic allocation problem in the real application and also for generated data. We believe that the lessons learned can be useful for other countries and for similar applications.

For future work we are planning to investigate the possibilities of improving the MILP model and applying other techniques, e.g. pre-processing, reducing the running time of the solution and making the optimisation approach feasible for larger applications.

Acknowledgements

We thank our colleagues, Dániel Horn and Zoltán Hermann for helping design the application on behalf for our Institute, Dávid Burka and his company for developing the website of the application (onkentesmentoralas.hu), and our collaborators in the two partner organisations, the Hungarian Reformed Church Aid and #school. Biró acknowledges the support by the Hungarian Scientific Research Fund – OTKA (No. K128611) and by the Hungarian Academy of Sciences, Momentum Grant no. LP2021-2.

References

Abraham, D. J., Blum, A., & Sandholm, T. (2007). Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM conference on electronic commerce, EC '07* (pp. 295–304). New York, NY, USA: ACM. https://doi.org/10.1145/1250910.1250954.

Agarwal, N., Ashlagi, I., Azevedo, E., Featherstone, C. R., & Karaduman, O. (2019). Market failure in kidney exchange. *American Economic Review, 109*, 4026–4070.

Agarwal, N., Ashlagi, I., Rees, M., Somaini, P., & Waldinger, D. (2021). Equilibrium allocations under alternative waitlist designs: Evidence from deceased donor kidneys. *Econometrica, 89*(1), 37–76.

Ágoston, K. C., Biró, P., & McBride, I. (2016). Integer programming methods for special college admissions problems. *Journal of Combinatorial Optimization, 32*, 1371–1399.

Ágoston, K. C., Biró, P., & Szántó, R. (2018). Stable project allocation under distributional constraints. *Operations Research Perspectives, 5*, 59–68.

Andersson, T., Ehlers, L., & Martinello, A. (2018). Dynamic refugee matching.

Ashlagi, I., & Roth, A. E. (2020). Kidney exchange: an operations perspective. Working paper.

Bansak, K., Ferwerda, J., Hainmueller, J., Dillon, A., Hangartner, D., Lawrence, D., & Weinstein, J. (2018). Improving refugee integration through data-driven algorithmic assignment. *Science, 359*, 325–329.

Biró, P. (2017). Applications of matching models under preferences.

Biró, P., Haase-Kromwijk, B., van de Klundert, J., et al., (2019). Building kidney exchange programmes in Europe – an overview of exchange practice and activities. *Transplantation, 103*, 1514–1522.

Biró, P., van de Klundert, J., Manlove, D., et al., (2021). Modelling and optimisation in European kidney exchange programmes. *European Journal of Operational Research, 291*, 447–456.

Biró, P., Manlove, D. F., & McBride, I. (2014). The hospitals/residents problem with couples: Complexity and integer programming models. In *International symposium on experimental algorithms* (pp. 10–21). Springer.

Biró, P., & McDermid, E. (2014). Matching with sizes (or scheduling with processing set restrictions). *Discrete Applied Mathematics, 164*, 61–67.

Bloch, F., Cantala, D., & Gibaja, D. (2020). Matching through institutions. *Games and Economic Behavior*.

Budish, E., Cachon, G. P., Kessler, J. B., & Othman, A. (2017). Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research, 65*, 314–336.

Cao, N. V., Fragniere, E., Gauthier, J.-A., Sapin, M., & Widmer, E. D. (2010). Optimizing the marriage market: An application of the linear assignment model. *European Journal of Operational Research, 202*, 547–553.

Delorme, M., García, S., Gondzio, J., Kalcsics, J., Manlove, D., & Pettersson, W. (2019). Mathematical models for stable matching problems with ties and incomplete lists. *European Journal of Operational Research, 277*, 426–441.

Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly, 69*, 9–15.

Garg, N., Kavitha, T., Kumar, A., Mehlhorn, K., & Mestre, J. (2010). Assigning papers to referees. *Algorithmica, 58*, 119–136.

Gerding, E., Perez-Diaz, A., Aziz, H., Gaspers, S., Marcu, A., Mattei, N., & Walsh, T. (2019). Fair online allocation of perishable goods and its application to electric vehicle charging.

Harris, D. N., Liu, L., Oliver, D., Balfe, C., Slaughter, S., & Mattei, N. (2020). How America's schools responded to the COVID crisis. *National Center for Research on Education Access and Choice & Education Research Alliance for New Orleans*. https://educationresearchalliancenola.org/files/publications/20200713-Technical-Report-Harris-etal-How-Americas-Schools-Responded-to-the-COVID-Crisis.pdf

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly, 2*, 83–97.

Kwanashie, A., & Manlove, D. F. (2014). An integer programming approach to the hospitals/residents problem with ties. In *Operations research proceedings 2013* (pp. 263–269). Springer.

Leshno, J. (2019). Dynamic matching in overloaded waiting lists. Available at SSRN 2967011.

Manlove, D. (2013). *Algorithmics of matching under preferences*: vol. 2. World Scientific.

Mattei, N., Saffidine, A., & Walsh, T. (2018). Fairness in deceased organ matching. In *Proceedings of the 2018 AAAI/ACM conference on AI, ethics, and society* (pp. 236–242).

Prendergast, C. (2016). The allocation of food to food banks. *EAI Endorsed Transactions on Serious Games, 3*, e4.

Roth, A. E., Sönmez, T., & Ünver, M. U. (2007). Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review, 97*, 828–851. https://doi.org/10.1257/aer.97.3.828.

Shimada, N., Yamazaki, N., & Takano, Y. (2020). Multi-objective optimization models for many-to-one matching problems. *Journal of Information Processing, 28*, 406–412.

So, M. C., Thomas, L. C., & Huang, B. (2016). Lending decisions with limits on capital available: The polygamous marriage problem. *European Journal of Operational Research, 249*, 407–416.

Trapp, A. C., Teytelboym, A., Martinello, A., Andersson, T., Ahani, N., et al., (2018). Placement optimization in refugee resettlement. *Technical report*.