European Journal of Operational Research 302 (2022) 50–61

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Stable matching of student-groups to dormitories

Nitsan Perach*, Shoshana Anily

Coller School of Management, Tel Aviv University, Tel Aviv 69978, Israel

ARTICLE INFO

Article history:
Received 18 February 2020
Accepted 30 December 2021
Available online 5 January 2022

Keywords:
Assignment
Dormitories
Stable matching
Many-to-many matching
Incentive compatibility

ABSTRACT

This paper generalizes results of former papers on the assignment of students to dormitories, under an entrance criterion, by allowing students to apply in groups. A group-application means that its applicants ask to be assigned to the same dormitory, and otherwise they prefer living off-campus. The underlying assumption in our model is that the dormitories share a common preference over the student-groups, which is given by a strictly increasing ranking of their credit scores. The definition of a quasi-stable outcome is adjusted in order to incorporate student-group applications, and we prove that such an outcome always exists. Furthermore, a polynomial-time algorithm that finds all the quasi-stable outcomes is proposed. Apparently, not all properties of the single students' model continue to hold under group-applications. Finally, we consider the incentive compatibility property of the proposed algorithm, and describe a specific quasi-stable outcome for which no subset of student-groups can gain by misrepresenting their preferences over the dormitories.

1. Introduction

Housing is a major factor in determining the quality of life that students experience during their college/university years. Providing students with on-campus good housing often results in making higher education institutions more attractive for potential students, and hence has been a major goal in the management of many of them (see https://www.spartnerships.com/colleges-and-universities-prefer-schools-with-attractive-student-housing-options/; Roche, Flanigan, & Copeland, 2010). It is often the case that on-campus housing in dormitories is available but supply is limited, and as a consequence the demand is higher than the supply. Moreover, there are many types of accommodations, which may differ in their location, size, quality, and fitness to special needs. See, for example, the detailed information on different types of dorms at the University of Illinois at Urbana Champaign (https://www.unigo.com/colleges/university-of-illinois-at-urbana-champaign/reviews/describe-the-dorms), which were designed to meet various students' preferences. Besides the room/apartment itself, a major factor that affects the quality of academic life of the students is the dormmates/roommates they are assigned with (see, for example Araujo & Murray, 2010; Khan, Shekili, Badi, & Khanbashi, 2020). In fact, the dorms' assignment is one of the considerations taken by students while choosing a university (see Khozaei, Hassan, & Razak, 2011; Reynolds, 2020 as well as the blogs on factors to consider while choosing a dormitory (https://www.grace.edu/10-things-to-consider-when-choosing-a-dorm/)).

In many campuses the implementation of the assignment mechanism is an important event that generates a lot of anxiety among students. Therefore, it becomes necessary to develop a matching method for assigning students to dormitories. At the Technion - Israel Institute of Technology (hereafter, abbreviated to the Technion), decisions about the assignment of students to dormitories are based on a three-step process. The first step determines eligibility for on-campus housing. The second step allocates the eligible students to dormitory categories, thereafter called dormitories, where each defines a certain type of accommodation. Finally, in the last step, students are assigned to specific rooms/apartments. Different criteria are used for these steps: socio-economic and personal data are the main factors for determining the merit score of each student for the first step, academic seniority and academic excellence are the main factors for determining the students' credit scores for the second step, and finally students' preferences over roommates, as for example, smoking/non-smoking or religious/no-religious partners, are taken into consideration in the third step.

In the three-step process described above, the preferences of the students over their roommates are considered only in the final (third) step. At all institutions of higher education that offer dorms, many appeals and withdrawals occur due to frustrated students that were not assigned with their friends to the same

* Corresponding author.
E-mail addresses: nitsan.perah@gmail.com (N. Perach), anily@tauex.tau.ac.il (S. Anily).

https://doi.org/10.1016/j.ejor.2021.12.048

dormitory. The process of forming groups of roommates is often not sufficiently transparent and might generate feelings of annoyance, as reflected by the following post Who gets quadded in Harvard housing (https://medium.com/harvard-open-data-project/harvard-housing-part-1-who-gets-quadded-a50221ae62c5). The objective of this paper is to generalize the current practice of submitting a separate housing application for any individual applicant, by allowing the submission of housing applications by general-sized groups of students, where the students of each such group ask to be assigned together to the same dormitory. More precisely, each applicant is included in exactly one group application, where a group consists of at least one student, meaning that applicants can also apply as singletons. A group-application should contain the individual socio-economic and academic achievements of each of its applicants, as well as a mutual ranking of a subset of the dormitories, which is acceptable by all members of the group. The housing department of the institution assigns upon the receipt of each group-application (i) a joint merit score, which is based on the respective socio-economic data of the members of the student-group, and (ii) a joint credit score, which is based on some aggregative measure of the academic scores of the students in the group.

There exist several techniques that generate a joint merit (credit) score for non-singleton groups based on the individual scores of its members, as for example, by using an order statistic of the individual scores like their minimum, or a certain percentile of the individual scores. We do not consider in this paper the techniques used for ranking the student-groups, so from now on we assume that the merit and credit scores of the student-groups are part of the input of our model.

We follow the paper of Perach et al. (see Perach, Polak, & Rothblum, 2007) that focuses on applications submitted by individuals to the housing department of the Technion. The paper incorporates an "entrance criterion", that is, a threshold value on the merit scores, where only applications whose merit score is not lower than the threshold value, are eligible for housing. The paper describes a matching algorithm that generates an assignment that has some desirable properties. In this paper, we consider, as in Perach et al. (2007), the case where all dormitories share a common ranking over the applications, which is based only on the credit scores associated with the applications.

In Perach & Rothblum (2010), the authors generalize the model and the algorithm proposed in Perach et al. (2007) to the case where the ranking of the dormitories over the individual students is not necessarily common. See Section 2 for more details.

Similarly to Perach et al. (2007), Perach & Rothblum (2010), this paper is based on the stable matching model of David Gale and Lloyd Shapley (GS) (see Gale & Shapley, 1962), hereafter called the classic matching model, or the GS model. The GS model defines a notion of stability for two-sided matching in populations where individuals have preferences over being matched with individuals of the other side. An algorithm that generates a stable matching in which no coalition of individuals can improve the fortunes of all of its members by switching their assignment with the assignment of other individuals, is proposed in Gale & Shapley (1962). Moreover, the output of the GS algorithm is optimal for one of the two populations that are matched, that is, each member of this population is assigned to his/her most favourable option among all other stable assignments. Thus, in the context of a dormitory assignment, a stable matching is expected to reduce the number of appeals submitted by students. From the point of view of the institution, a lower number of appeals submitted to the housing department, signifies a higher level of satisfaction by the applicants, which is an important factor in the goodwill of the institute.

Some other implementations with similar traits to the matching problem of student-groups to dormitories include:

(1) The assignment of students to graduate business schools, where students are required to take a GMAT exam (see Azar & Siwei, 2014): the GMAT scores can serve as the current merit scores, while the work experience of candidates, which is a major determinant in the admission process, can serve as the credit scores in our model.
(2) The assignment of families registered for apartments in a global lottery (see, for example, the Buyer's Price Program (https://www.gov.il/en/service/request_for_eligibility_confirmation) in Israel, and the NYC housing lottery (https://www.nytimes.com/2020/06/15/nyregion/nyc-affordable-housing-lottery.html)). In such applications, the lottery generates the merit scores, while the registration date can serve as the credit scores (early registration results in a higher score).
(3) The assignment of people infected by the Covid-19 virus to Corona hotels (see, for example, the program of BC Housing (https://www.bchousing.org/publications/COVID-19-Isolation-Hotels-Fact-Sheet.pdf)). Infected people staying in such hotels are positive to the virus but they do not need special treatment (not like the ones in hospitals). As everyone in the hotel is positive, people there are not required to be in quarantine and they can be assigned to a room with other people. Infected people are eligible for accommodation in a Corona hotel if some of their family members living with them at the same house are negative to the virus, and the house is not suitable for separating the healthy family members from the infected ones, which are required to be in quarantine. The merit scores of the infected people are based on the unsuitability of their house to separate them from the non-infected people living at the same house. The hotel ranks the infected people that are qualified for a Corona hotel accommodation according to the proximity of the hotel to their residence.

In this paper we prove that a stable matching of student-groups to dormitories, exists. The paper is organized as follows: A literature review on related matching problems is described in Section 2. In Section 3 we present the group assignment model discussed here. Section 4 presents some properties of group assignment referring to the classic matching model, where dormitories' preferences over the student-groups are common. Section 5 describes the properties of the groups' assignment in a matching model which incorporates an entrance criterion: a stable assignment for any instance is shown to exist and an algorithm that generates all possible stable assignments, is proposed. In addition, we elaborate on the properties of the assignment of individual students, discussed in Perach et al. (2007), Perach & Rothblum (2010), and verify which properties continue to hold for the student-group matching model. In Section 6, we discuss the existence of the incentive compatibility property for any set of student- groups, namely, the property that it is impossible for all the student-groups in the set to gain by misrepresenting their preference-lists. In Section 7 we allow ties in the common ranking of the dormitories over the student-groups, and discuss some of the properties of this generalized model. Section 8 includes some conclusions and comments. Algorithms and examples are deferred to the Appendix, which is concluded in Section A.2.2 by a simulation study of the algorithm proposed in Section 5: we use real data on the 11 Technions' dormitories and their forecast for the number of single students' housing applications for the academic year 2021/22. Using this forecast, we randomly form the student-groups and their attributes, and then apply the algorithm and analyse its output.

2. Literature review

Various applications of the stable matching problem are described in the literature, as for example:

(1) The assignment of students to schools: students' satisfaction has improved upon replacing the existing assignment mechanism by the one that generates a student-optimal GS stable assignment (see Abdulkadiroglu, Pathak, & Roth, 2005a; Abdulkadiroglu, Pathak, & Roth, 2005b; Abdulkadiroglu & Sonmez, 2003; Balinski & Sõnmez, 1999).
(2) The assignment of interns to hospitals, where hospitals compete over interns, and each intern has preferences over the hospitals (see Roth, 1984; Roth, 1996; Roth, 2003).
(3) Kidney transplants from live donors: the number of successful transplants has greatly increased by implementing a variant of the GS algorithm (see Roth, Sonmez, & Unver, 2007). Some practical matching models are described in Biró et al. (2021).

The classic GS matching model and its extensions involve diverse types of mathematical analysis (see, for example, Knuth, 1976; Manlove, 2013; Roth & Sotomayor, 1990).

This paper is closely related to Perach et al. (2007), which proposes a stable matching model for the problem of assigning individual students to dormitories at the Technion. The proposed matching model incorporates an entrance criterion: each applicant is assigned a unique merit score based on her personal socio-economic data. A threshold value on the merit scores is determined according to the total capacity of the dormitories. The eligible applicants are the ones whose merit score is above this threshold value. The ranking of the applicants by the dormitories is based on their credit scores, and hence is common and complete. Each applicant submits a preference-list over a subset of dormitories. The actual assignment of eligible students takes into account the students' preferences over the dormitories as well as the common preference-list of the dormitories over the applicants. A stable assignment in this model, fulfills at least one of the following two conditions: either all the dormitories are at full capacity, or all applicants are found eligible. The paper proposes an algorithm that generates, for any instance, a stable matching that satisfies some desirable properties.

The framework of Perach et al. (2007) is generalized in Perach & Rothblum (2010), by relaxing the assumption of a common and complete preference-list of the dormitories over the applicants. In particular, each dormitory is allowed to use its own evaluation criteria for stating its preference-list over the applicants. The proposed algorithm for the generalized model preserves most of the properties of the original algorithm. Moreover, Perach & Rothblum (2010) demonstrates that the outcome of the new algorithm satisfies the incentive compatibility property for a single student, which means that a single applicant cannot improve her assignment by misrepresenting her preference-list while all other applicants state their true preference-lists.

In this paper, we extend the above student assignment model to allow for group applications, where the students of any group ask to be assigned to the same dormitory. The group assignment problem has initially been studied in Roth (1984) in the context of matching between residents and hospitals, motivated by the need to assign not only individual residents but also couples of residents who want to be assigned to the same hospital or to nearby hospitals. Each individual resident submits a preference ranking over the hospitals and each couple submits a preference ranking over pairs of hospitals. Finally, each hospital ranks the residents as individuals. The paper proves that a stable assignment does not necessarily exist. Moreover, the problem of deciding whether there exists a stable assignment for a given instance has been proved to be NP-complete even if there are no single residents and each hospital has only one position (see Ronn, 1990). Several papers point out special cases where a stable assignment exists (see, for example, Cantala, 2004; Dutta & Massó, 1997; Klaus & Klijn, 2005; Klaus & Klijn, 2007; Kojima, Parag, & Roth, 2013 and references therein). The natural restriction where the preference-list of any couple is consistent with the preferences of its members is discussed in McDermid & Manlove (2010): the paper provides a polynomial time algorithm that finds a stable assignment or reports that none exists.

Other applications of group-assignment include:

(1) The matching problem of children to schools, see Ashlagi & Shi (2014), where the input of the model is a non-strict ranking of children over the schools and vice versa. The objective is to assign as many children as possible from the same neighbourhood to the same school in order to strengthen the community cohesion in schools.
(2) The problem of assigning students to projects under constraints is discussed in Chiarandini, Fagerberg, & Gualandi (2019): students rank project-topics and the goal is to assign students to groups and groups to projects, while satisfying side constraints in a fair way.

3. Preliminaries and notations

In this section we present a modification of the stable matching model with an entrance criterion (as presented in Perach et al., 2007; Perach & Rothblum, 2010), which incorporates "group applications".

The data for our model includes two disjoint finite sets G and D, referred to as the set of student-groups and the set of dormitories, respectively. Let $|G| = n$, $|D| = k$, $G = \{g_1, \ldots, g_n\}$ and $D = \{d_1, \ldots, d_k\}$. Note that any applicant belongs to a single group $g \in G$, which is possibly a singleton. Each student-group $g \in G$ is associated with three nonnegative independent numbers $q_g \in \mathbf{N}$, $m_g \in \Re^+$ and $c_g \in \Re^+$, where q_g is the size of the group, m_g is its merit score, and c_g is its credit score. The special case where $\sum_{g\in G} q_g = n$ refers to singleton student-groups, which is considered in Perach et al. (2007). Throughout the paper we assume that:

(1) Each of the sequences $m_{g_1}, \ldots, m_{g_n}$ and $c_{g_1}, \ldots, c_{g_n}$ consists of distinct numbers, that is, for $g \neq g'$: $m_g \neq m_{g'}$ and $c_g \neq c_{g'}$.
(2) W.l.o.g, student-groups in G are indexed in a decreasing order of their credit scores.

In addition, each student-group $g \in G$ is associated with a non-empty set $\emptyset \subset D_g \subseteq D$, and a ranking $\succ_g$ of $D_g \cup \{g\}$, where g is the least preferred element by student-group g. We refer to $\succ_g$ as the preference of student-group g over the set of dormitories in D_g and over being unassigned to any dormitory in its preference-list, where the latter case is represented by g. We say that dormitory $d \in D$ is *acceptable* by student-group $g \in G$ if and only if $d \in D_g$. Similarly, $d \notin D_g$ means that student-group $g \in G$ finds dormitory $d \in D$ *unacceptable* and it prefers to be assigned to g, which means living off-campus, rather than living in dormitory d. The ranking of the student-groups by the dormitories is common and complete, and is based on the groups' credit scores. Each dormitory is associated with a capacity, which is the number of beds that it offers. However, in order to avoid the case of having unnecessary beds that no student-group is interested in, we consider the "effective capacity" b_d of each dormitory d, to be the minimum between the number of beds in d and $\sum_{\{g\in G|d\in D_g\}} q_g$, where the second term stands for the total number of students that accept dormitory d. For simplicity, in the sequel we assume that the effective capacity of a dormitory equals its given capacity. In addition, we assume that for any student-group $g \in G$ and any dormitory $d \in D_g$, $q_g \leq b_d$.

An *assignment* μ of a set $A \subseteq G$ over the dormitories in D is a set of pairs, $\mu = \{(g,d) | g \in A, d \in D_g\}$, where (i) the student-groups in different pairs are distinct, and (ii) $\sum_{g:(g,d)\in\mu} q_g \le b_d$ for each $d \in D$. Assignment μ can also be interpreted as a function $\mu(\cdot): A \to D$ having $\mu(g) = d$ for $g \in A$. Under this interpretation, μ represents the assignment of student-groups in the set A to dormitories, while the student-groups in $G \backslash A$ are not assigned.

An *outcome* is a triplet (μ, W, R) where μ is the assignment of the student-groups of a set $A \subseteq G$, while the student-groups that are not assigned are partitioned into two disjoint sets W and R, namely, $G\backslash A = W \cup R$: the student-groups in the set W are called *waiting* student-groups, while those in R are called *refugees*.[1] The set W contains student-groups that currently are not considered for a dormitory assignment because their merit scores are not sufficiently high, while the student-groups in the complement set $G\backslash W = (A \cup R)$ have all been considered: each student-group in A has already been assigned to a dormitory in its preference-list, where for any student-group in R, no dormitory in its preference-list with sufficient vacancy is left.

Next, we present a definition of plausibility that generalizes the corresponding definition in Perach et al. (2007), Perach & Rothblum (2010) for student-groups of size one.

Definition 1. An outcome (μ, W, R) is said to be <u>plausible</u> if:

(a) $|W| \ge 1$ implies that $m_g < m_{g'}$ for any two student-groups $g \in W$ and $g' \in G\backslash W$;
(b) either $W = \emptyset$ or $\sum_{d\in D} b_d - \sum_{(g,d)\in\mu} q_g < q_{\hat{g}}$, where $\hat{g}$ is the student-group whose merit score is the highest among the student-groups in W.

Condition (a) of Definition 1 asserts that the maximum merit score of a student-group in W is strictly lower than the minimum merit score of a student-group in the complement set, namely $G\backslash W$. For a given outcome (μ, W, R), the only student-groups that are not considered by the assignment process are the ones in the set W.

It follows immediately that if (μ, W, R) and (μ', W', R') are two outcomes that satisfy condition (a) of Definition 1, then W and W' are ordered by set inclusion.[2] Condition (b) of Definition 1 asserts that either all student-groups are processed, or the total number of free beds that are left in all dormitories is smaller than the size of the student-group with the highest merit score among the student-groups in W.

Note that Definition 1 has not made any use of the credit scores, which next play a central role in the determination of the final outcome.

Definition 2. A pair $(g,d) \in (G\backslash W) \times D$ is a <u>blocking pair</u> of an outcome (μ, W, R) if the following conditions hold:

(a) $d \in D_g$.
(b) $g \in R$ or $d \succ_g \mu(g)$ (which implies $(g,d) \notin \mu$).
(c) $q_g + \sum_{(i,d)\in\mu} q_i - \sum_{i\in G'_g} q_i \le b_d$, where $G'_g = \{g' | (g',d) \in \mu \text{ and } c_g > c_{g'}\}$.

In other words, according to Definition 2, a blocking pair of a specific outcome consists of a student-group that is not in the set of waiting student-groups, and a dormitory such that the two are not assigned one to the other, though each of them prefers being assigned to the other rather than to its current state in the outcome. Note that item (c) implies that after removing from dormitory d student-groups whose credit scores are lower than the credit score of student-group g, the dormitory has sufficient vacancy for student-group g.

Definition 3. An outcome (μ, W, R) is said to be <u>internally stable</u> if no blocking pairs exist, and it is said to be <u>quasi-stable</u> if it is plausible and internally stable.

The following example demonstrates the four possible combinations regarding plausibility and internal stability of outcomes.

Example 1. Let $G = \{g_1, g_2, g_3, g_4\}$, $D = \{d_1, d_2\}$, $q_{g_1} = 2, q_{g_2} = q_{g_3} = q_{g_4} = 1$ and $b_{d_1} = b_{d_2} = 2$. Recall that $c_{g_i} > c_{g_j}$ for $i < j$. Suppose that $m_{g_3} > m_{g_4} > m_{g_1} > m_{g_2}$ and each student-group of G prefers dormitory d_1 over dormitory d_2. Consider the following four outcomes:

- $(\mu_1, W_1, R_1) = (\{(g_1,d_2),(g_2,d_1)\}, \{g_3,g_4\}, \emptyset)$: this outcome is not internally stable as the pair (g_1, d_1) blocks it. In addition it is not plausible as there is a free bed in dormitory d_1, while the highest merit scored student-group in W_1, namely g_3, is singleton.
- $(\mu_2, W_2, R_2) = (\{(g_1,d_2),(g_3,d_1),(g_4,d_1)\}, \{g_2\}, \emptyset)$: this outcome is not internally stable as the pair (g_1, d_1) blocks it. However, this outcome satisfies both conditions of plausibility.
- $(\mu_3, W_3, R_3) = (\{(g_1,d_1),(g_3,d_2),(g_2,d_2)\}, \{g_4\}, \emptyset)$: this outcome is internally stable, but it is not plausible. Student-group $g_3 \in W_3$ has a higher merit score than student-group g_2, which is not in W_3.
- $(\mu_4, W_4, R_4) = (\{(g_1,d_1),(g_3,d_1),(g_4,d_1)\}, \{g_2\}, \emptyset)$: this outcome is internally stable and plausible, and hence it is quasi-stable. ○

As a side remark note that if all the student-groups were of equal size $q \ge 2$, then this size could be scaled down to 1 while scaling down by q also the capacities of all the dormitories. This would have generated a model that is equivalent to the one with single students, a case that has been considered in Perach et al. (2007), Perach & Rothblum (2010). Thus, here we assume that the student-groups are of non-identical general size.

4. A stable assignment for a given *W*

This section presents some properties of internally stable outcomes. We first fix a set W of waiting student-groups that consists of the x $(0 \le x \le n)$ student-groups whose merit-scores are the lowest among the n merit-scores of the student-groups in G. We then restrict ourselves to the subset $G' \subset G$ that consists of the student-groups that are not in W. In what follows, the pair (G', D) is referred to as a *market*.

In the sequel we prove that for any given market (G', D) there exists a unique internally stable outcome. The proof starts by presenting a constructive algorithm, called the SD(G) algorithm, that follows the Serial Dictatorship (SD) principles (see Abdulkadiroglu & Sonmez, 1998), and is an adaptation of the Greedy-SMI-1ML algorithm (see Section 2 of Irving, Manlove, & Scott, 2008) for student-group assignment. Recall that the student-groups are indexed in a decreasing order of their credit scores. The algorithm scans the student-groups of G' according to an increasing order of their indices, and assigns each student-group its most preferred dormitory that still has enough beds to accommodate it. If such a dormitory does not exist, the student-group is classified as a refugee.

The algorithm calls the function $f(\hat{\succ}_{g_i})$, which returns the current most preferred dormitory of student-group g_i, that is, the first element in the preference-list $\hat{\succ}_{g_i}$. If such a dormitory does not exist, then $f(\hat{\succ}_{g_i}) = \emptyset$.

The algorithm uses the following data-structure:

- g_i - the current student-group scanned by the algorithm.

[1] The term refugees is used to emphasize the uncertainty of the status of these applicants. This term has also been used in earlier publications of the first author, see Perach et al. (2007), Perach & Rothblum (2010).

[2] That is, exactly one of the following holds: (1) $W = W'$ (2) $W \subset W'$, or (3) $W' \subset W$.

- $\hat{\succ}_{g_i}$ - the current preference-list of student-group g_i, which contains the dormitories in D_{g_i} that have not yet rejected it.
- $d_{\hat{\succ}_{g_i}}$ - the most preferred dormitory in $\hat{\succ}_{g_i}$.
- c - a Boolean variable that signifies if the current student-group of G' has been assigned to a dormitory.
- R - the current set of unassigned student-groups of G', which could not be assigned to a dormitory in their preference-list.
- μ - the current assignment of student-groups in $G'\backslash R$ to dormitories.

The output (μ, R) consists of the assignment μ, which is a set of pairs $(g, d) \in G' \times D$ that are matched, and a set $R \subseteq G'$ that consists of the refugee student-groups that could not be assigned by the algorithm. Note that all students in $G'\backslash R$ are assigned a dormitory under μ.

Algorithm 1: Serial Dictatorship for groups (SD(G)) algorithm.

Input: market (G', D) where $G' \neq \emptyset$
Output: (μ, R), where μ is an assignment, and R is a refugee set

$\mu \leftarrow \emptyset, R \leftarrow \emptyset, i \leftarrow 1, c \leftarrow'' true''$;
while $i < n + 1$ **do**
  if $g_i \in G'$ **then**
    $\hat{\succ}_{g_i} \leftarrow \succ_{g_i}$;
    $c \leftarrow'' false''$
  $d_{\hat{\succ}_{g_i}} \leftarrow f(\hat{\succ}_{g_i})$;
  while $d_{\hat{\succ}_{g_i}} \neq \emptyset$ *and* $c ='' false''$ **do**
    if $b_{d_{\hat{\succ}_{g_i}}} - \sum_{\{j|(g_j, d_{\hat{\succ}_{g_i}}) \in \mu\}} q_{g_j} \geq q_{g_i}$ **then**
      $\mu \leftarrow \mu \cup \{(g_i, d_{\hat{\succ}_{g_i}})\}$;
      $c \leftarrow'' true''$
    else
      remove dormitory $d_{\hat{\succ}_{g_i}}$ from $\hat{\succ}_{g_i}$;
      $d_{\hat{\succ}_{g_i}} \leftarrow f(\hat{\succ}_{g_i})$
  if $c ='' false''$ **then**
    $R \leftarrow R \cup \{g_i\}$
  $i \leftarrow i + 1$
output (μ, R)

Comment 1: The complexity of the SD(G) algorithm is of order $O(\sum_{g \in G'} |D_g|)$ as each student-group in G' and its preference-list are scanned at most once. Thus, the complexity is $O(|G'|k)$, where $k = |D|$, as the maximum length of the preference-list of each student-group is bounded by the number of dormitories.

The following theorem provides a characterization of the outcome generated by the SD(G) algorithm. The theorem extends Theorem 2.1 of Irving et al. (2008) for the student-group assignment:

Theorem 1. *For any market (G', D), $G' \subseteq G$, there exists a single internally stable outcome of the form $(\mu, G\backslash G', R)$.*

Proof. We start by showing that for any market (G', D), $G' \subseteq G$, the SD(G) algorithm terminates and generates an internally stable outcome of the form $(\mu, G\backslash G', R)$. Thereafter, we show uniqueness. Termination of the algorithm is immediate as each student-group is scanned exactly once, and the preference-list of each student-group is of bounded length. Let $(\mu, G\backslash G', R)$ be the outcome generated by the SD(G) algorithm on market (G', D). Note that in assignment μ each student-group is assigned to a dormitory only if the dormitory is acceptable by the student-group and it has enough beds to accommodate it. Now, assume by contradiction that the pair (g, d) is a blocking pair in the outcome $(\mu, G\backslash G', R)$. Consider the time of addressing student-group g during the algorithm. At this time, all dormitories in D contain only student-groups whose credit scores are higher than c_g. If at this point of time, there were enough beds to accommodate student-group g in a dormitory $d' \neq d$, which student-group g prefers better than d, then student-group g would be assigned to a dormitory that it prefers better than d, and therefore the pair (g, d) will not be a blocking-pair as item (b) of Definition 2 is not satisfied. Otherwise, if d had enough beds to accommodate g, then the pair (g, d) would be a member of assignment μ, and again the pair (g, d) will not be a blocking pair. Finally, if dormitory d did not have enough beds to accommodate student-group g, then the pair (g, d) would not be a blocking pair as item (c) of Definition 2 is not satisfied, namely the student-groups that have been assigned to dormitory d up to this point of time, have higher credit scores than c_g, as according to the SD(G) algorithm no student-group leaves a dormitory once it was assigned to it.

Next, we show that the outcome $(\mu, G\backslash G', R)$ is a unique internally stable outcome for market (G', D). Also this part is proved by contradiction: assume that in a given market (G', D) there were two different internally stable outcomes $(\mu_1, G\backslash G', R_1)$ and $(\mu_2, G\backslash G', R_2)$. W.l.o.g., let $g^* \in G'$ be the student-group with the highest credit score such that $\mu_1(g^*) \succ_{g^*} \mu_2(g^*)$ or $g^* \in R_2\backslash R_1$, implying that $g^* \notin R_1$. Thus, there exists a dormitory $d_1 \in D$ such that:

$$d_1 = \mu_1(g^*) \succ_{g^*} \mu_2(g^*) \quad or \quad g^* \in R_2 \tag{1}$$

Consider the outcome $(\mu_2, G\backslash G', R_2)$: let $\tilde{G}_{g^*} = \{g' | \mu_2(g') = d_1 \text{ and } c_{g^*} < c_{g'}\}$ be the set of student-groups that have been assigned by μ_2 to dormitory d_1 and their credit scores are higher than the credit score of student-group g^*. As student-group g^* is the highest credit-scored student-group whose assignment differs between the outcomes $(\mu_1, G\backslash G', R_1)$ and $(\mu_2, G\backslash G', R_2)$, and since student-group g^* is assigned to dormitory d_1 under μ_1, clearly,

$$q_{g^*} + \sum_{i \in \tilde{G}_{g^*}} q_i \leq b_{d_1}. \tag{2}$$

Note that (1) and (2) imply that the pair (g^*, d_1) blocks outcome $(\mu_2, G\backslash G', R_2)$, in contradiction to the assumption on the internal stability of $(\mu_2, G\backslash G', R_2)$. □

To conclude this section, we propose for the group assignment of market (G', D), a variant of the GS algorithm that has been proposed for matching two sets of individuals (see Gale & Shapley, 1962; Roth & Sotomayor, 1990). The student-group assignment version of the GS algorithm, thereafter denoted by GS(G), starts by having all the student-groups of G' unassigned, and all beds of D are free. In each step of the algorithm, each student-group approaches its most preferred dormitory, that has not yet rejected it. As a consequence, the new assignment of each dormitory consists of the maximum number of the highest credit scored student-groups among those that have approached it and those that were assigned to it in the previous assignment, without exceeding its capacity. Student-groups that were previously assigned to a dormitory but are excluded from the new assignment, are considered as being rejected by this dormitory. The algorithm reiterates until each student-group is assigned to a certain dormitory or is rejected by all of them. A student-group that is rejected by all the dormitories in its preference-list, is classified as a refugee. For a more detailed algorithm and its complexity, see the Appendix.

The following example demonstrates a market (G, D), where the output of the SD(G) algorithm and the output of the GS(G) algorithm do not coincide. Moreover, the outcome $(\mu_{GS(G)}, \emptyset, R_{GS(G)})$, where $(\mu_{GS(G)}, R_{GS(G)})$ is the output of the GS(G) algorithm on that market, is not necessarily internally stable. An illustration of the algorithms on the following example is presented in Section A.1 of the Appendix.

Example 2. Let $G = \{g_1, g_2, g_3, g_4\}$, $D = \{d_1, d_2\}$, $q_{g_1} = q_{g_3} = 2, q_{g_2} = q_{g_4} = 1$ and $b_{d_1} = b_{d_2} = 2$. Recall that $c_{g_i} > c_{g_j}$ for $i < j$. The preference-lists of the student-groups are given in the following table:

g_1	g_2	g_3	g_4
d_2	d_2	d_1	d_1
d_1	d_1		d_2

The output of the SD(G) algorithm on market (G, D) is $(\mu_{SD(G)}, R_{SD(G)})$, where $\mu_{SD(G)} = \{(g_1, d_2), (g_2, d_1), (g_4, d_1)\}$ and $R_{SD(G)} = \{g_3\}$. The output of the GS(G) algorithm on market (G, D) is $(\mu_{GS(G)}, R_{GS(G)})$, where $\mu_{GS(G)} = \{(g_1, d_2), (g_2, d_1)\}$ and $R_{GS(G)} = \{g_3, g_4\}$.

Thus, $(\mu_{GS(G)}, R_{GS(G)}) \neq (\mu_{SD(G)}, R_{SD(G)})$. Moreover, the outcome $(\mu_{GS(G)}, \emptyset, R_{GS(G)})$ is not internally stable, as the pair (g_4, d_1) blocks it (dormitory d_1 has an additional free bed under this outcome). ◦

5. Quasi-stability

The previous section has determined the unique internally stable assignment under a certain market. In this section, we consider the general model with merit scores, and elaborate on some of its properties. We first note that unlike the model with single students (see Section 3 in Perach et al., 2007 and Perach & Rothblum, 2010), in the general-sized student-groups' model there may exist outcomes, (μ, W, R) and (μ', W', R'), $W' \subset W$, which fulfill condition (a) of plausibility, where (μ, W, R) is plausible, but (μ', W', R') is not.

Example 3. Let $G = \{g_1, g_2, g_3\}$, $D = \{d_1\}$, $q_{g_1} = q_{g_2} = 1, q_{g_3} = 2$, $b_{d_1} = 2$, and $m_{g_1} < m_{g_2} < m_{g_3}$. Assume that dormitory d_1 is acceptable by all the student-groups, and recall that $c_{g_1} > c_{g_2} > c_{g_3}$. Consider the following outcomes:

- (μ, W, R), where $\mu = \{(g_3, d_1)\}$, $W = \{g_1, g_2\}$ and $R = \emptyset$.
- (μ', W', R'), where $\mu' = \{(g_2, d_1)\}$, $W' = \{g_1\}$ and $R' = \{g_3\}$.

Clearly, (μ, W, R) is a quasi-stable outcome, but (μ', W', R') is not as it does not satisfy condition (b) of Definition 1. ◦

In Section A.2 of the Appendix, we propose an algorithm that finds all the quasi-stable outcomes for any given sets G and D. In sub-Section 5.1 we shortly describe the idea behind the algorithm, and some of its properties. In Sections 5.2 and 5.3, we focus on a specific quasi-stable outcome and elaborate on some of its properties.

5.1. Finding all quasi-stable outcomes

An algorithm that finds all stable matchings for the classic matching model, is presented in Gustfield & Irving (1989). In Section A.2 we present an algorithm, called the QSO algorithm, that finds all quasi-stable outcomes for a given set of student-groups G and a given set of dormitories D, and discuss its complexity. More specifically, for any possible set of waiting student-groups that satisfies condition (a) of plausibility, the algorithm finds an internally stable outcome, by running the SD(G) algorithm while filtering out outcomes that do not satisfy condition (b) of plausibility. A simulation study of the QSO algorithm over real data obtained from the Technions' dormitory management, is presented in Section A.2.2.

The QSO algorithm starts with an empty set H of quasi-stable outcomes. It generates a run of the SD(G) algorithm on all markets of the form (G', D), where G' consists of the y, $1 \leq y \leq n$, student-groups in G whose merit scores are the highest. More precisely, let $G' = \emptyset$. For $y = 1 \ldots n$ the algorithm does the following: it removes the highest merit-scored student-group from G and inserts it into G'. Then if the output of the SD(G) algorithm on market (G', D), where $|G'| = y$, satisfies condition (b) of Definition 1, the respective outcome with $W' = G \backslash G'$, $|W'| = n - y$, is inserted into the set H of quasi-stable outcomes. Thereafter, as long as $y < n$, the highest merit scored student-group in W' is transferred to G', and the SD(G) algorithm is called again. The last iteration of the algorithm is, in fact, the run of the SD(G) algorithm on market (G, D), that is, when $y = n$ and the set W of waiting student-groups is empty.

The following lemma characterizes the output of the QSO algorithm:

Lemma 1.

1. *The output H of the QSO algorithm presented in Section A.2 of the Appendix, is equal to the set of all quasi-stable outcomes for the set of student-groups G and the set of dormitories D.*
2. *The complexity of the QSO algorithm is of order $O(n \sum_{g \in G} |D_g|)$, or equivalently $O(n^2 k)$.*

Proof.

1. By definition, any triplet $(\mu_h, W_h, R_h) \in H$ generated by the QSO algorithm is an outcome, and by Theorem 1, any such outcome is internally stable. Plausibility holds due to the way that the algorithm determines the sets of waiting student-groups. The proof that the output H consists of all the quasi-stable outcomes follows immediately from Theorem 1, and the fact that the QSO algorithm considers all possible sets of waiting student-groups for the given data.
2. According to Comment 1, each run of the SD(G) algorithm on market (G', D) is of complexity $O(\sum_{g \in G'} |D_g|)$ or $O(nk)$. The QSO algorithm on G and D considers all possible sets of waiting student-groups, where a set of waiting student-groups is either empty or it contains a proper subset of student-groups of G whose merit scores are the lowest, implying that condition (a) of plausibility holds. There is a total of n such possibilities. For each set of waiting student-groups, the QSO algorithm generates a run of the SD(G) algorithm. Therefore, the complexity of the QSO algorithm is of order $O(n \sum_{g \in G} |D_g|)$ or, equivalently, $O(n^2 k)$.

□

Note that the outcome that is considered in the last iteration of the QSO algorithm is the result of running the SD(G) algorithm on market (G, D), whose set of waiting student-groups is empty. By Theorem 1, such an outcome exists, is unique, and it satisfies plausibility as its set of waiting student-groups is empty. The following observation, is therefore, straightforward:

Observation 1. For any sets G and D, there exists at least one quasi-stable outcome, namely the outcome associated with $W = \emptyset$, implying that the set H of quasi-stable outcomes is non-empty, that is $|H| \geq 1$.

5.2. Properties of quasi-stable outcomes

Several optimality criteria may be useful in evaluating quasi-stable outcomes. For example, the one with the least number of refugee student-groups, the one with the least number of refugee individual students, the one with the largest number of student-groups (or individual students) that get assigned to their first choice dormitory, the one that minimizes the unused capacity (number of free beds), etc. Note that finding a quasi-stable outcome under any possible criterion can be done by scanning the set H generated by the QSO algorithm.

We elaborate here on some properties of the first outcome generated by the QSO algorithm, denoted by (μ_1, W_1, R_1). In the special case where all the student-groups are singletons, this outcome coincides with the outcome of Dor-AA in Perach et al. (2007), and all optimality criteria stated above, do hold.

The following claim states that outcome the outcome (μ_1, W_1, R_1) has the largest number of waiting student-groups among all quasi-stable outcomes, and therefore the largest number of waiting students as individuals. As a consequence, R_1 contains the least number of students. To this end we introduce the function $s : 2^G \to \{1, \ldots, \sum_{i=1}^{n} q_i - n + 1\}$, which returns the number of students in any subset G' of G.

Claim 1. *Let H be the output of the QSO algorithm on the sets of student-groups G and dormitories D, and suppose that $|H| > 1$. Consider the outcome $(\mu_1, W_1, R_1) \in H$. For any other outcome $(\mu_j, W_j, R_j) \in H$, $j > 1$, the following properties hold:*

1. *$W_j \subset W_1$ (which implies that $s(W_j) < s(W_1)$).*
2. *$s(R_j) > s(R_1)$.*

Proof. The proof of the first part follows immediately from the way the set H is generated by the QSO algorithm. In order to prove the second part, let e_k be the total number of free beds in outcome (μ_k, W_k, R_k), that is, $e_k = \sum_{d \in D} b_d - \sum_{(g,d) \in \mu_k} q_g$ for $k = 1, j$. Therefore, $s(R_k) = \sum_{g \in G \backslash W_k} q_g - (\sum_{d \in D} b_d - e_k)$ for $k = 1, j$. In view of the first part, $G \backslash W_1 \subset G \backslash W_j$. Let g^* be the student-group with the highest merit score in W_1. Condition (b) of plausibility asserts that $q_{g^*} > e_1$. By the definition of the QSO algorithm, $g^* \in G \backslash W_j$. Thus,

$$s(R_j) - s(R_1) = \sum_{g \in G \backslash W_j} q_g - \sum_{g \in G \backslash W_1} q_g + e_j - e_1 \geq q_{g^*} + e_j - e_1$$
$$\geq q_{g^*} - e_1 > 0 \tag{3}$$

concluding the proof. □

Note also that Claim 1 guarantees that R_1 has the smallest number of students, but this is not necessarily achieved by the smallest number of student-groups, as demonstrated by the following example.

Example 4. Let $G = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9\}$, $D = \{d_1, d_2, d_3, d_4\}$, $q_{g_i} = 1$ for $i \in \{1, 3, 4, 5, 8, 9\}$, $q_{g_i} = 2$ for $i \in \{2, 6, 7\}$ and $b_{d_\ell} = 2$ for $1 \leq \ell \leq 4$. Let $m_{g_9} > m_{g_8} > m_{g_7} > m_{g_6} > m_{g_3} > m_{g_2} > m_{g_4} > m_{g_5} > m_{g_1}$. The preference-lists of the student-groups are given in the following table:

g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
d_1	d_1	d_1	d_2	d_2	d_3	d_4	d_3	d_4
	d_2		d_3	d_4				

The following two outcomes are quasi-stable: the outcome (μ_1, W_1, R_1) generated by the QSO algorithm, where $\mu_1 = \{(g_2, d_1), (g_4, d_2), (g_5, d_2), (g_6, d_3), (g_7, d_4)\}$, $W_1 = \{g_1\}$, and $R_1 = \{g_3, g_8, g_9\}$, and the outcome (μ_2, W_2, R_2), where $\mu_2 = \{(g_1, d_1), (g_2, d_2), (g_3, d_1), (g_4, d_3), (g_5, d_4), (g_8, d_3), (g_9, d_4)\}$, $W_2 = \emptyset$ and $R_2 = \{g_6, g_7\}$. Thus, in the outcome (μ_1, W_1, R_1) there are three student-groups which are determined to be refugees, where in the outcome $(\mu_2, W_2 = \emptyset, R_2)$ only two student-groups are determined to be refugees. ◦

5.3. Optimality criteria

In view of Claim 1, in the set of outcomes H, student-groups in $G \backslash W_1$ are not members of any set of waiting student-groups W_j, $j = 1, \ldots, |H|$. In other words, each student-group in $G \backslash W_1$ has a sufficiently high merit score, which guarantees that it will be processed by the SD(G) algorithm in all iterations of the QSO Algorithm. Thus, we refer to the student-groups (students) in $G \backslash W_1$ as *needy* student-groups (needy students).

5.3.1. Optimality for student-groups and individual students

In this sub-section we first propose a definition of optimality for needy student-groups, that is those that belong to $G \backslash W_1$, see Claim 1:

Definition 4. A quasi-stable outcome is said to be optimal for a needy student-group g if g is not assigned by any other quasi-stable outcome to a dormitory that it prefers better[3] according to its preference-list.

Next we prove that outcome (μ_1, W_1, R_1) is optimal for at least one student-group.

Theorem 2. *Let $\breve{g} \in G$ be the student-group with the highest credit score in $G \backslash W_1$. Then, outcome (μ_1, W_1, R_1) is optimal for $\breve{g}$.*

Proof. The proof is immediate since student-group $\breve{g}$ is scanned first during the run of the SD(G) algorithm when generating outcome (μ_1, W_1, R_1). Therefore, student-group $\breve{g}$ is assigned under (μ_1, W_1, R_1) to its most preferred dormitory. □

Next, we demonstrate by an example that unlike the case of singleton student-groups (see Perach et al., 2007; Perach & Rothblum, 2010), for general sized student-groups there may be cases where no outcome that is optimal for all the needy student-groups exists.

Example 5. Let $G = \{g_1, g_2, g_3\}$, $D = \{d_1, d_2\}$, $q_{g_1} = q_{g_3} = 1, q_{g_2} = 2$, and $b_{d_1} = 1, b_{d_2} = 2$. Let $m_{g_3} > m_{g_2} > m_{g_1}$. The following table presents the preference-lists of the student-groups:

g_1	g_2	g_3
d_2	d_2	d_2
d_1		d_1

The QSO algorithm generates two quasi-stable outcomes. The first is (μ_1, W_1, R_1), where $\mu_1 = \{(g_2, d_2), (g_3, d_1)\}$, $W_1 = \{g_1\}$, implying that the set of needy student-groups is $\{g_2, g_3\}$. Furthermore, $R_1 = \emptyset$. The second quasi-stable outcome is (μ_2, W_2, R_2), where $\mu_2 = \{(g_1, d_2), (g_3, d_2)\}$, $W_2 = \emptyset$ and $R_2 = \{g_2\}$. By Definition 4, we consider the optimality of an outcome only for the needy student-groups: Outcome (μ_1, W_1, R_1) is optimal for student-group g_2, while outcome (μ_2, W_2, R_2) is optimal for student-group g_3. ◦

In addition, unlike the case where all student-groups are singletons (discussed in Perach et al., 2007; Perach & Rothblum, 2010), the following example shows that there may exist a quasi-stable outcome $(\mu_j, W_j, R_j) \neq (\mu_1, W_1, R_1)$, in which most needy students, and also most needy student-groups, prefer their assignment under (μ_j, W_j, R_j) over their assignment under outcome (μ_1, W_1, R_1).

Example 6. Let $G = \{g_1, g_2, g_3, g_4, g_5\}$, $D = \{d_1, d_2, d_3, d_4\}$, $q_{g_2} = 2, q_{g_i} = 1$ for $i \in \{1, 3, 4, 5\}$ and $b_{d_1} = 2, b_{d_2} = b_{d_3} = b_{d_4} = 1$. Let $m_{g_2} > m_{g_3} > m_{g_4} > m_{g_5} > m_{g_1}$. The following table presents the preference-lists of the student-groups:

g_1	g_2	g_3	g_4	g_5
d_1	d_1	d_1	d_1	d_1
d_2		d_2	d_2	d_2
d_3		d_3	d_3	d_3
d_4		d_4	d_4	d_4

The first outcome generated by the QSO algorithm is (μ_1, W_1, R_1), where $\mu_1 = \{(g_2, d_1), (g_3, d_2), (g_4, d_3), (g_5, d_4)\}$, $W_1 = \{g_1\}$ and $R_1 = \emptyset$, implying that the set of needy student-groups $G \backslash W_1 = \{g_2, g_3, g_4, g_5\}$. The quasi-stable outcome (μ_2, W_2, R_2) is generated for $W_2 = \emptyset$, where $\mu_2 =$

[3] Definition 4 includes the preference of being assigned to a dormitory over being a refugee.

$\{(g_1,d_1),(g_3,d_1),(g_4,d_2),(g_5,d_3)\}$, and $R_2=\{g_2\}$. Note that outcome (μ_2,W_2,R_2) is strictly better than outcome (μ_1,W_1,R_1) for the three students in student-groups $g_3,g_4,g_5\in G\backslash W_1$, while for student-group g_2 that consists of two students, the opposite holds. Thus, we conclude that outcome (μ_2,W_2,R_2) is optimal for more needy students and for more needy student-groups than outcome (μ_1,W_1,R_1). ∘

5.3.2. Optimality for a student-group whose preference-list is complete

In the case of singleton student-groups it is known that a student-group that lists all dormitories as acceptable will never end up as a refugee student-group under (μ_1,W_1,R_1), see Theorem 1(e) in Perach et al. (2007). It turns out that this claim does not hold for general-sized student-groups: in fact, a singleton student-group that lists all dormitories as acceptable might end up as a refugee in outcome (μ_1,W_1,R_1).

Example 7. Let $G=\{g_1,g_2,g_3,g_4\}$, $D=\{d_1,d_2\}$, $q_{g_2}=2, q_{g_1}=q_{g_3}=q_{g_4}=1$ and $b_{d_1}=b_{d_2}=2$. Let $m_{g_4}>m_{g_3}>m_{g_2}>m_{g_1}$. The following table presents the preference-lists of the student-groups:

g_1	g_2	g_3	g_4
d_1	d_1	d_1	d_1
d_2	d_2		d_2

Consider the only quasi-stable outcome for this data, namely (μ_1,W_1,R_1), where $\mu_1=\{(g_1,d_1),(g_3,d_1),(g_2,d_2)\}$, $W_1=\emptyset$ and $R_1=\{g_4\}$. Thus, all dormitories are acceptable by student-group g_4, but $g_4\in R_1$. ∘

6. Incentive compatibility

Incentive compatibility addresses the question of whether each member of a group of agents can gain by misrepresenting its preference-list, while all other agents state their true preference-lists. It is well known that in the implementation of the men courting version of the GS algorithm in the classic stable matching model (see Gale & Shapley, 1962), such a manipulation is not profitable to any group of men. That is, there exists at least one man in this group that is matched to the same woman or alternatively prefers his match under the true preference-lists than his match under the misrepresentation of the preference-lists, see Dubins & Friedman (1981), and Theorem 4 in Gale & Sotomayor (1985). An immediate conclusion which follows is that such a manipulation is not profitable when applying the GS algorithm in the many-to-one matching models, as in the assignment of residents to hospitals, since any hospital with p open positions can be presented as p hospitals, each having a single open position.

In the singleton student-groups model, the SD(G) applied on a market (G,D) satisfies the incentive compatibility property (see Theorem 1 of Svensson (1999)). The following theorem demonstrates that this property holds also for the general-sized student-groups case.

Theorem 3. *Let $\succ_G=(\succ_{g_1},\ldots,\succ_{g_n})$ be the vector of true preference-lists of the student-groups in market (G,D). Let $\tilde{\succ}_G=(\tilde{\succ}_{g_1},\ldots,\tilde{\succ}_{g_n})$ be a vector of preference-lists that coincides with the vector $\succ_G$ except for a non-empty subset of student-groups $\hat{G}\subseteq G$. Finally, let (μ,R) and $(\tilde{\mu},\tilde{R})$ be the assignments generated by application of the SD(G) algorithm on market (G,D) under $\succ_G$ and under $\tilde{\succ}_G$, respectively. Then, there exists a student-group $g\in\hat{G}$, for which exactly one of the following holds: (1) $\mu(g)\succ_g\tilde{\mu}(g)$ or (2) $\mu(g)=\tilde{\mu}(g)$ or (3) $g\in\tilde{R}$.*

Proof. Let $g_i\in\hat{G}$ be the student-group whose credit score is the highest in $\hat{G}$. Note that the scanning order of the student-groups of G by the SD(G) algorithm on market (G,D) depends only on the credit scores and it is independent of the student-groups' preference-lists. Consider the time when student-group g_i is scanned by the SD(G) algorithm. The temporary assignments at this specific point of time under $\succ_G$ and under $\tilde{\succ}_G$ coincide, as for all $j<i$, $\succ_{g_j}=\tilde{\succ}_{g_j}$. Thus, at this point of time, either student-group g_i is assigned the best dormitory that still has a sufficient number of beds to host it according to its true preference-list, and if none exists it will be determined as a refugee. Clearly, student-group g_i cannot do better by misrepresenting its preference-list, as at this point of time all the higher ranked options, if such ones exist, are non-feasible. The proof follows as during the run of the SD(G) algorithm no student-group is being removed from a dormitory. □

Theorem 3 refers to internally stable outcomes for market (G,D). Recall that the last iteration of the QSO algorithm is applied on market (G,D) and the resulting outcome is, by definition, a quasi-stable outcome, since the respective set W of waiting student-groups is empty. This outcome is, in fact, the last outcome added to the set H of all quasi-stable outcomes generated by the QSO algorithm.

Next, we refer to the first outcome in the set H generated by the QSO algorithm, namely (μ_1,W_1,R_1). Consider the respective algorithm, called the 1-QSO algorithm, which boils down to the first iteration of the QSO algorithm. According to Theorem 2 of Perach & Rothblum (2010), when applying the 1-QSO algorithm on singleton student-groups, no student-group can be assigned to a dormitory that it prefers better if it misrepresented its preferences, while all other student-groups stated their true preference-lists.[4] The following example shows that this property does not hold in the general-sized student-groups case, i.e., a student-group can be assigned by the 1-QSO algorithm to a better preferred dormitory by misrepresenting its preference-list.

Example 8. Consider Example 6. The first outcome of the QSO algorithm is (μ_1,W_1,R_1), where $\mu_1=\{(g_2,d_1),(g_3,d_2),(g_4,d_3),(g_5,d_4)\}$, $W_1=\{g_1\}$, and $R_1=\emptyset$. In particular, student group g_5 is assigned to d_4, which is its least preferred dormitory in its original preference-list.

Consider the following modification of the original preference-lists, with respect to student-group g_5:

g_1	g_2	g_3	g_4	g_5
d_1	d_1	d_1	d_1	d_1
d_2		d_2	d_2	d_2
d_3		d_3	d_3	d_3
d_4		d_4	d_4	

The first outcome of the QSO algorithm is $(\tilde{\mu}_1,\tilde{W}_1,\tilde{R}_1)$, where $\tilde{\mu}_1=\{(g_1,d_1),(g_3,d_1),(g_4,d_2),(g_5,d_3)\}$, $\tilde{W}_1=\emptyset$ and $\tilde{R}_1=\{g_2\}$. Thus, by manipulating the preference-lists, student-group g_5 is assigned to d_3 under outcome $(\tilde{\mu}_1,\tilde{W}_1,\tilde{R}_1)$, which is better preferred than its assignment to d_4 under outcome (μ_1,W_1,R_1), when the true preference-lists are used. ∘

Example 8 demonstrates not only that a student-group can get a better assignment for itself by misrepresenting its preference-list, but that the assignment may be better for other student-groups as well. Moreover, in this example, all the students, except the ones in student-group g_2, gain from this "lie", implying that lying is beneficial for most of the students in G. Example 8 also shows that a "lie" of a single student-group can influence the size of the set of waiting student-groups of the first outcome generated by the QSO algorithm.

7. General credit scores

In the previous sections, the credit scores of the student-groups were assumed to be distinct. As this assumption does not necessar-

[4] For the singleton student-group case, the 1-QSO algorithm coincides with G-DorAA, which is presented in Perach & Rothblum (2010).

ily hold in practice, we consider here the general case. Given the input of n student-groups, let $c_1 > \ldots > c_r$, $1 \le r < n$[5], be the r distinct values that the credit scores assume. Let G^{c_i}, for $i = 1, \ldots, r$, be the *credit sets*, which are non-empty sets of students that share the same credit score c_i, and $G = \cup_{i=1}^{r} G^{c_i}$. The student-groups in each set G^{c_i}, for $1 \le i \le r$, are indexed consecutively and arbitrarily from $|\cup_{j=1}^{i-1} G^{c_j}| + 1$ to $|\cup_{j=1}^{i} G^{c_j}|$. Any such assignment of the indices to the student-groups in each credit set $G^{c_i} \in G$ forms a *permutation* of the student-groups over G. Let P^G be the set of all permutations over G, that is, $|P^G| = \prod_{i=1}^{r} |G^{c_i}|!$. In addition, for any subset $G' \subseteq G$, let $P^{G'}$ be the subset of permutations of the student-groups of G', that is $|P^{G'}| = \prod_{i=1}^{r} |G^{c_i} \cap G'|!$. Next, we consider the following procedure, which is an adaptation of the SD(G) algorithm for general credit scores.

Procedure Random Serial Dictatorship for groups (Rand-SD(G))

Input: market (G', D) where $G' \neq \emptyset$, anda random permutation $p \in P^{G'}$

Output: (μ, R), where μ is an assignment, and R is a refugee set

Call the SD(G) algorithm where the student-groups are indexed according to p, and return its output.

The following theorem provides a characterization of the outcomes generated by the Rand-SD(G) procedure.

Theorem 4. *For any market (G', D), $G' \subseteq G$, there exist at most $|P^{G'}|$ internally stable outcomes of the form $(\mu, G \backslash G', R)$.*

Proof. According to Theorem 1, any call to the SD(G) algorithm generates a unique internally stable outcome for market (G', D). Hence, the Rand-SD(G) procedure generates an internally stable outcome for any permutation $p \in P^{G'}$. As there exist $|P^{G'}|$ permutations that refer to the student-groups of G', the number of internally stable outcomes for that market is bounded by $|P^{G'}|$. □

The following observation follows immediately from Theorem 3.

Observation 2. The Rand-SD(G) procedure applied on market (G', D) satisfies the incentive compatibility property for any subset of student-groups that misrepresent their preference-lists.

Recall that the QSO algorithm generates all the quasi-stable outcomes for G and D, where the credit scores are distinct. The following ND-QSO procedure calls the QSO algorithm in order to generate all the quasi-stable outcomes for general credit scores. We assume here that the elements of P^G are indexed arbitrarily.

Procedure QSO for general credit scores (ND-QSO)

Input: $G \neq \emptyset$ and $D \neq \emptyset$

Output: a set H of quasi-stable outcomes

1 **for** $i = 1$ *to* $|P^G|$ **do**

2 $\quad H_i \leftarrow$ QSO(G, D) where the student-groups are indexed according to permutation p_i

3 Return $H = \bigcup_{j=1}^{|P^G|} H_i$

Lemma 2.

1. *The output H of the ND-QSO procedure is equal to the set of all quasi-stable outcomes for the set of student-groups G and the set of dormitories D.*
2. *The complexity of the ND-QSO procedure is of order $O(|P^G| * n * \sum_{g \in G} |D_g|)$ or, equivalently, $O(|P^G| * n^2 k)$.*

Proof. The proof of both parts follows immediately from Lemma 1 and the fact that the algorithm checks every possible permutation. □

8. Conclusions and comments

This paper extends the stable matching model with an entrance criterion to include applications from groups of students who prefer living off-campus over being separated in different dormitories. We adjust the definition of quasi-stable outcomes, explore the properties of the new model and develop algorithms that generate outcomes with certain desirable properties. In addition, we show that some of the properties that hold for the singleton student-groups model, continue to hold for the generalized model. Finally, the existence of the incentive compatibility property in the generalized model is discussed.

From a social point of view, the quasi-stable outcomes generated by the QSO algorithm are not equivalent. The larger is the set of waiting student-groups in the outcome, the more sensitive is the respective assignment to the merit scores. The last outcome, in which the set of waiting student-groups is empty, totally ignores the merit scores and only credit scores are taken into consideration in the assignment process. This last outcome is therefore optimal for the dormitories, but it is the worst from a socio-economic point of view, as the merit scores do not play any role in generating the assignment.

The model presented here refers to a situation where the dormitories' preferences, which are represented by the credit scores of the student-groups, are common and complete. The existence of common preferences guarantees that if a student-group is not acceptable by a certain dormitory, then it is not acceptable by any other dormitory. Such a student group can be excluded from further consideration at the beginning of the assignment process, as in any quasi-stable outcome the unacceptable student-groups will either be in the set of waiting student-groups or will end up as refugees.

Acknowledgements

The research of the authors has been supported by the following grants of the second author: the Israel Science Foundation, grants no. 338/15 and 1489/19, the Coller Foundation and the Henry Crown Israeli Institute for Business Research.

The authors wish to thank the dormitory management of the Technion - Institute of Technology for their cooperation in providing us with the data, and the Ph.D. candidate Guy Kern for helping with the simulations. In addition, the authors wish to thank the Associate Editor and the anonymous reviewers for their helpful comments.

Appendix A.

A1. The assignment of student-groups by the GS(G) algorithm

Recall that the function $f(\succ_g)$ returns the most preferred dormitory by student-group g. Define the function $l(d, \mu)$ to return the lowest credit-scored student-group that is assigned to dormitory d under assignment μ. The algorithm uses the following data structure:

- P - the current set of student-groups that have not been assigned yet.
- R - the current set of refugees.

[5] The case where all credit scores are distinct has been considered in the former sections.

Algorithm 2: The Gale Shapley algorithm for student-groups (GS(G)).

Input: market (G, D), where $G \neq \emptyset$
Output: assignment μ
$\mu \leftarrow \emptyset,\ R \leftarrow \emptyset,\ P \leftarrow G,\ fin = ''\ false''$;
while $P = \emptyset$ **do**
  for $i = 1$ *to* n **do**
    if $g_i \in P$ **then**
      $P \leftarrow P \backslash \{g_i\}$;
      $d_{g_i} \leftarrow f(\succ_{g_i})$;
      $\mu \leftarrow \mu \cup \{(g_i, d_{g_i})\}$
  for $j = 1$ *to* $|D|$ **do**
    while $b_{d_j} < \sum_{\{g:(g,d_j)\in\mu\}} q_g$ **do**
      $\bar{g} \leftarrow l(d_j, \mu)$;
      remove d_j from $\succ_{\bar{g}}$;
      $\mu \leftarrow \mu \backslash \{(\bar{g}, d_j)\}$;
      if $\succ_{\bar{g}} = \emptyset$ **then**
        $R \leftarrow R \cup \{\bar{g}\}$
      else
        $P \leftarrow P \cup \{\bar{g}\}$
output μ

- μ - the current assignment of student-groups of $G \backslash R$ to dormitories.

The complexity of the GS(G) algorithm is of order $O(\sum_{i \in G'} |D_i|)$, as all the student-groups of G may approach all the dormitories in D. Thus, the complexity can also be written as $O(nk)$, where n is the number of student-groups, and k is the number of dormitories.

A1.1. Illustration of the SD(G) and GS(G) algorithms

In the following we execute the algorithms on Example 2.

The SD(G) algorithm:

i	μ	R	$\bar{\succ}_{g_i}$	$d(g_i)$
1	$\emptyset$	$\emptyset$	d_2, d_1	d_2
1	$\{(g_1, d_2)\}$	$\emptyset$	d_2, d_1	d_2
2	$\{(g_1, d_2)\}$	$\emptyset$	d_2, d_1	d_2
2	$\{(g_1, d_2)\}$	$\emptyset$	d_1	d_1
2	$\{(g_1, d_2), (g_2, d_1)\}$	$\emptyset$	d_1	d_1
3	$\{(g_1, d_2), (g_2, d_1)\}$	$\emptyset$	d_1	d_1
3	$\{(g_1, d_2), (g_2, d_1)\}$	$\emptyset$	$\emptyset$	$\emptyset$
3	$\{(g_1, d_2), (g_2, d_1)\}$	$\{g_3\}$	$\emptyset$	$\emptyset$
4	$\{(g_1, d_2), (g_2, d_1)\}$	$\{g_3\}$	d_1, d_2	d_1
4	$\{(g_1, d_2), (g_2, d_1), (g_4, d_1)\}$	$\{g_3\}$	d_1, d_2	d_1

The GS(G) algorithm:

Current step	P	μ	R	g'	$d_{g'}$	i	$\bar{g}$
initialization	$\{g_1, g_2, g_3, g_4\}$	$\emptyset$	$\emptyset$	-	-	-	-
1	$\{g_1, g_2, g_3\}$	$\{(g_1, d_2)\}$	$\emptyset$	g_1	d_2	-	-
1	$\{g_1, g_2\}$	$\{(g_1, d_2), (g_2, d_2)\}$	$\emptyset$	g_2	d_2	-	-
1	$\{g_1\}$	$\{(g_1, d_2), (g_2, d_2), (g_3, d_1)\}$	$\emptyset$	g_3	d_1	-	-
1	$\emptyset$	$\{(g_1, d_2), (g_2, d_2), (g_3, d_1), (g_4, d_1)\}$	$\emptyset$	g_4	d_1	-	-
2	$\{g_4\}$	$\{(g_1, d_2), (g_2, d_2), (g_3, d_1)\}$	$\emptyset$	-	-	1	g_4
2	$\{g_2, g_4\}$	$\{(g_1, d_2), (g_3, d_1)\}$	$\emptyset$	-	-	2	g_2
1	$\{g_4\}$	$\{(g_1, d_2), (g_2, d_1), (g_3, d_1)\}$	$\emptyset$	g_2	d_1	-	-
1	$\emptyset$	$\{(g_1, d_2), (g_2, d_1), (g_3, d_1), (g_4, d_2)\}$	$\emptyset$	g_4	d_2	-	-
2	$\emptyset$	$\{(g_1, d_2), (g_2, d_1), (g_4, d_2)\}$	$\{g_3\}$	-	-	1	g_3
2	$\emptyset$	$\{(g_1, d_2), (g_2, d_1)\}$	$\{g_3, g_4\}$	-	-	2	g_4

A2. The QSO algorithm

The QSO Algorithm finds all the quasi-stable outcomes. Define the function $m(A)$ that returns the student-group in a set $A \subseteq G$ with the highest merit score. The following data structure is used by the algorithm:

- $\hat{W}$ - the current set of waiting student-groups.
- H - the up-to-date set of quasi-stable outcomes.
- h - the index of the last quasi-stable outcome inserted into H.
- $G' = G \backslash \hat{W}$.
- (μ', R') - the current assignment that is generated by the SD(G) algorithm.

Algorithm 3: The QSO algorithm.

Input: $G \neq \emptyset$ and $D \neq \emptyset$
Output: a set H of quasi-stable outcomes
$g^* \leftarrow m(G)$;
$\hat{W} \leftarrow G \backslash \{g^*\}$;
$H \leftarrow \emptyset, h \leftarrow 1, G' \leftarrow \{g^*\}, itr \leftarrow 1$;
while $itr \leq n$ **do**
  $(\mu', R') \leftarrow$ SD(G)(G', D) ;
  $g^* \leftarrow m(\hat{W})$;
  if $\hat{W} = \emptyset$ *or* $\sum_{d\in D} b_d - \sum_{(g,d)\in\mu'} q_g < q_{g^*}$ **then**
    $(\mu_h, W_h, R_h) \leftarrow (\mu', \hat{W}, R')$;
    $H \leftarrow H \cup \{(\mu_h, W_h, R_h)\}$;
    $h \leftarrow h + 1$
  $\hat{W} \leftarrow \hat{W} \backslash g^*$;
  $G' \leftarrow G' \cup \{g^*\}$;
  $itr \leftarrow itr + 1$
output H

A2.1. Illustration of the QSO algorithm

In this subsection we execute the QSO algorithm on Example 6.

itr	H	h	G′	μ'	R′	$\hat{W}$
1	$\emptyset$	1	$\{g_2\}$	-	-	$g_3 \to g_4 \to g_5 \to g_1$
1	$\emptyset$	1	$\{g_2\}$	$\{(g_2, d_1)\}$	$\emptyset$	$g_3 \to g_4 \to g_5 \to g_1$
2	$\emptyset$	1	$\{g_2, g_3\}$	$\{(g_2, d_1)\}$	$\emptyset$	$g_4 \to g_5 \to g_1$
2	$\emptyset$	1	$\{g_2, g_3\}$	$\{(g_2, d_1), (g_3, d_2)\}$	$\emptyset$	$g_4 \to g_5 \to g_1$
3	$\emptyset$	1	$\{g_2, g_3, g_4\}$	$\{(g_2, d_1), (g_3, d_2)\}$	$\emptyset$	$g_5 \to g_1$
3	$\emptyset$	1	$\{g_2, g_3, g_4\}$	$\{(g_2, d_1), (g_3, d_2), (g_4, d_3)\}$	$\emptyset$	$g_5 \to g_1$
4	$\emptyset$	1	$\{g_2, g_3, g_4, g_5\}$	$\{(g_2, d_1), (g_3, d_2), (g_4, d_3)\}$	$\emptyset$	g_1
4	$\emptyset$	1	$\{g_2, g_3, g_4, g_5\}$	$\{(g_2, d_1), (g_3, d_2), (g_4, d_3), (g_5, d_4)\}$	$\emptyset$	g_1
5	$\{(\mu_1, W_1, R_1) = (\{(g_2, d_1), (g_3, d_2), (g_4, d_3), (g_5, d_4)\}, \{g_1\}, \emptyset)\}$	2	$\{g_1, g_2, g_3, g_4, g_5\}$	$\{((g_1, d_1), (g_3, d_1), (g_4, d_3), (g_5, d_4)\}$	$\emptyset$	*null*
5	$\{(\mu_1, W_1, R_1) = (\{(g_2, d_1), (g_3, d_2), (g_4, d_3), (g_5, d_4)\}, \{g_1\}, \emptyset)\}$	2	$\{g_1, g_2, g_3, g_4, g_5\}$	$\{((g_1, d_1), (g_3, d_1), (g_4, d_2), (g_5, d_3)\}$	$\{g_2\}$	*null*
6	$\{(\mu_1, W_1, R_1) = (\{(g_2, d_1), (g_3, d_2), (g_4, d_3), (g_5, d_4)\}, \{g_1\}, \emptyset), \{(\mu_2, W_2, R_2) = (\{((g_1, d_1), (g_3, d_1), (g_4, d_2), (g_5, d_3)\}, \emptyset, \{g_2\})\}$	3	$\{g_1, g_2, g_3, g_4, g_5\}$	$\{((g_1, d_1), (g_3, d_1), (g_4, d_2), (g_5, d_3)\}$	$\{g_2\}$	*null*

A2.2. Simulation of the QSO algorithm on real data

In this subsection we report the results of a simulation study of the QSO algorithm over real data obtained from the dormitory management of the Technion for the academic year 2021–2022. The data refers to dormitories that host only single students of both genders. The assignment process of couples of students at the

Table 1
Dormitories' capacity.

Dormitory	Number of beds
Lower dormitories	589
Renovated dormitories	119
Nations dormitories	144
Senate dormitories	248
Neve America dormitories	289
Shilon dormitories	11
Palm Beach dormitories	15
Undergraduate dormitories	466
Canada dormitories	806
Old East dormitories	214
New East dormitories	770
Total	**3671**

Table 2
Frequency table of the number of quasi-stable outcomes in the 103 iterations.

Number of quasi-stable outcomes	97	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
Number of iterations	1	1	5	3	8	8	7	9	7	11	14	5	8	2	3	2	2	1	1	3	2

Table 3
Frequency of the number of refugee-groups.

Number of refugee student-groups	1	2	3	4
Number of iterations	76	24	2	1

Technion is simpler as all of them are assigned to another dormitory with enough capacity.

The data consists of 4000 applications of students and 11 dormitories. Table 1 presents the number of beds in each dormitory. As our model refers to group applications, where the data refers to single students, we randomly aggregate individual students into groups, as described below.

We simulate the algorithm by using the *R* language. The simulation study consists of 103 iterations of the QSO algorithm. In each iteration we first generate the student-groups and their characteristics, and conclude by running the QSO algorithm on the respective input. The average number of student-groups in a single iteration of the QSO algorithm is about 1300, implying that this is the number of times that the SD algorithm is run per iteration of the QSO algorithm. The average running time of a single iteration of the QSO algorithm is about 20 min, implying that the average time of running a single iteration of the SD algorithm is $\frac{20*60}{1300} < 1$ second. We next describe the way of generating the data for each iteration.

For $i = 1, \dots, 103$ do begin:

1. Generate student-groups of size bounded by 5, with a total of 4000 students, by drawing independent and identically distributed random variables from a discrete uniform distribution on $\{1, \dots, 5\}$, until the total size of the student-groups is at least 3995. The size of the last group is determined so that the total size of the student-groups is 4000. Let γ_i be the number of student-groups generated in iteration i.
2. Generate two independent complete rankings over the γ_i student-groups: one represents their ranking according to the merit scores and the other their ranking according to the credit scores. The student-groups are then indexed in a decreasing order of their credit scores from 1 up to γ_i.
3. For each student-group $g, 1 \le g \le \gamma_i$, generate a complete preference-list over the 11 dormitories as follows: the dormitory at the top of the preference-list of student-group g is drawn from a discrete uniform distribution on $\{1, \dots, 11\}$. Denote it by d_g^1. The next dormitory in the preference-list is uniformly drawn from the ten integers $\{1, \dots, 11\} \backslash \{d_g^1\}$, etc. The last dormitory in the preference-list is the only dormitory that has not been drawn previously.
4. The complete preference-lists of the student-groups, generated in Step 3, are independently and randomly trimmed from the bottom so that all dormitories that were trimmed are considered as unacceptable by the respective student-group. We assume that the probability mass function of the length of a preference-list is increasing in between 1 and 11, as most of the student-groups will tend to accept most (or even all) dormitories, in order to increase their chance to be assigned a dormitory. Hence, in order to trim a given preference-list from its bottom, we let X be a random variable distributed according to a truncated geometric distribution over the integers in $\{1, \dots, 11\}$, where the size of the respective preference-list is distributed according to $12 - X$. The implementation is done by applying the following two steps:
   (a) Generate a random number θ_i from the continuous uniform distribution on (0,1), which represents the fraction of student-groups in iteration i that consider the last dormitory in their current preference-list as unacceptable.
   (b) For each student-group g, $1 \le g \le \gamma_i$, draw independently a number from the uniform distribution in (0,1). If the number is smaller than θ_i then remove the last dormitory from the current preference-list. This step is repeated until either the number drawn is at least θ_i, or the current preference-list consists of a single dormitory.
5. Apply the QSO algorithm on the data generated by steps 1 − 4 and the dormitories in Table 1. Let H_i be the corresponding output.

End.

Next, we analyse the number of quasi-stable outcomes obtained in each iteration, namely the size of the set H_i for $i = 1, \dots, 103$. Table 2 presents the frequency of the number of quasi-stable outcomes for the 103 iterations of the QSO algorithm.

As Table 2 demonstrates, the two most frequent outcomes are 109 and 110. A possible explanation for this observation is that the dormitories contain a total of 3671 beds, where the number of applications is 4000, namely, at least 329 students are not assigned, independently of the assignment process. The average group size in our simulation is 3. Thus, approximately $\frac{329}{3} \in [109, 110]$ student-groups are in the largest possible waiting-list, namely W_1. Moreover, it is most probable that after finding the first quasi-stable outcome, namely (μ_1, W_1, R_1), most other triplets considered by the QSO algorithm do not violate plausibility, and hence they are also quasi-stable outcomes.

Next we consider the refugee set R_1 of the first outcome generated in each of the 103 runs of the simulation. The existence of refugee students, whose merit score justifies the need to assign them a dormitory, but no acceptable dormitory can host them, has raised concerns within the management of the dormitories at the Technion (see Perach et al., 2007). The following two frequency tables show the number of refugee student-groups and the number refugee students as individuals over the 103 iterations of the QSO algorithm, respectively.

Tables 3 and 4 show that on average there are 1.3 refugee student-groups and 4.4 refugee students. These numbers are negligible relatively to the number of beds and the number of applications.

Table 4
Frequency of the number of refugee students.

Number of refugee students	1	2	3	4	5	6	7	8	9	10	11
Number of iterations	7	5	20	23	33	1	5	4	3	1	1

References

Abdulkadiroglu, A., Pathak, P. A., & Roth, A. E. (2005a). The new york city high school match. *American Economic Review, 95*, 364–367.

Abdulkadiroglu, A., Pathak, P. A., & Roth, A. E. (2005b). The boston public school match. *American Economic Review, 95*, 368–371.

Abdulkadiroglu, A., & Sonmez, T. (1998). Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica, 66*(3), 689–701.

Abdulkadiroglu, A., & Sonmez, T. (2003). School choice: A mechanism design approach. *American Economic Review, 93*, 729–747.

Araujo, P., & Murray, J. (2010). Estimating the effects of dormitory living on student performance. *Economics Bulletin, 30*, 866–878.

Ashlagi, A., & Shi, P. (2014). Improving community cohesion in school choice via correlated-lottery implementation. *Operations Research, 62*(6), 1247–1264.

Azar, A., & Siwei, C. (2014). Stability and strategy-proofness for college admissions with an eligibility criterion. *Review of Economic Design, 19*, 47–66.

Balinski, M., & Sõnmez, T. (1999). A tale of two mechanisms: Student placement. *Journal of Economic Theory, 84*, 73–94.

Biró, P., de Klundert, J. V., Manlove, D., Pettersson, W., Andersson, T., Burnapp, L., … Viana, A. (2021). Modelling and optimization in european kidney exchange programs. *European Journal of Operational Research, 291*, 447–456.

Cantala, D. (2004). Matching markets: The particular case of couples. *Economic Bulletin, 3*(45), 1–11.

Chiarandini, M., Fagerberg, R., & Gualandi, S. (2019). Handling preferences in student-project allocation. *Annals of Operations Research, 275*(1), 39–78.

Dubins, L. E., & Friedman, D. A. (1981). Machiavelli and the Gale–Shapley algorithm. *American Mathematical Monthly, 88*, 485–494.

Dutta, B., & Massó, J. (1997). Stability of matchings when individuals have preferences over colleagues. *Journal of Economic Theory, 75*, 464–475.

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

Gale, D., & Sotomayor, M. (1985). Some remarks on the stable matching problem. *Discrete Applied Mathematics, 11*, 223–232.

Gustfield, D., & Irving, R. W. (1989). *The stable marriage problem: Structure and algorithms*. Cambridge: The MIT Press.

Irving, R. W., Manlove, D. F., & Scott, S. (2008). The stable marriage problem with master preference lists. *Discrete Applied Mathematics, 156*(15), 2959–2977.

Khan, F., Shekili, N., Badi, A., & Khanbashi, H. (2020). Exploring the impact of hostel life of students on academic performance: Sohar university - a case study. *International Journal of Research in Entrepreneurship and Business Studies, 1*, 1–14.

Khozaei, F., Hassan, A., & Razak, N. (2011). Development and validation of the student accommodation preferences instrument (SAPI). *Journal of Building Appraisal, 6*, 299–313.

Klaus, B., & Klijn, F. (2005). Stable matchings and preferences of couples. *Journal of Economic Theory, 121*, 75–106.

Klaus, B., & Klijn, F. (2007). Paths to stability for matching markets with couples. *Games and Economic Behaviour, 58*, 154–171.

Knuth, D. E. (1976). Marriages stables. *Les Presses de UdeM*.

Kojima, F., Parag, A., & Roth, A. E. (2013). Matching with couples: Stability and incentives in large markets. *Quarterly Journal of Economics, 128*(4), 1585–1632.

Manlove, D. F. (2013). *Algorithmics of matchings under preferences. Series on theoretical computer science (vol. 2)*. World Scientific Publishing.

McDermid, E. J., & Manlove, D. F. (2010). Keeping partners together: Algorithmic results for the hospitals/residents problem with couples. *Journal of Combinatorial Optimization, 19*, 279–303.

Perach, N., Polak, J., & Rothblum, U. G. (2007). A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the technion. *International Journal of Game Theory, 10*, 519–535.

Perach, N., & Rothblum, U. G. (2010). Incentive compatibility for the stable matching model with an entrance criterion. *International Journal of Game Theory, 39*, 657–667.

Reynolds, C. (2020). The effect of dormitory residence during college on student outcomes. *Journal of Human Capital, 14*(2), 249–289.

Roche, C., Flanigan, M., & Copeland, J. R. (2010). Student housing: Trends, preferences and needs. *Contemporary Issues in Education Research, 3*, 45.

Ronn, E. (1990). NP-Complete stable matching problems. *Journal of Algorithms, 11*, 285–304.

Roth, A. E. (1984). The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy, 92*, 991–1016.

Roth, A. E. (1996). The national resident matching program as a labour market. *Journal of the American Medical Association, 275*(13), 1054–1056.

Roth, A. E. (2003). The origins, history, and design of the resident match. *Journal of the American Medical Association, 289*, 909–912.

Roth, A. E., Sonmez, T., & Unver, M. U. (2007). Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review, 97*, 828–851.

Roth, A. E., & Sotomayor, M. (1990). Two-sided matching: A study in game-theoretic modeling and analysis. *Econometric Society Monograph*.

Svensson, L. G. (1999). Strategy-proof allocation of indivisible goods. *Social Choice and Welfare, 16*(4), 557–567.