Contents

AN ANALYSIS OF A REAL-LIFE ALLOCATION PROBLEM

by

MAHMUT KEMAL ÖZBEK

Submitted to the Social Sciences Institute
in partial fulfillment of the requirements for the degree of
Master of Arts

Sabancı University
June 2007

AN ANALYSIS OF A REAL-LIFE ALLOCATION PROBLEM

APPROVED BY

Assist. Prof. Dr. Özgür KIBRIS ....
(Thesis Supervisor)

Prof. Dr. Mehmet BAÇ ....

Assist. Prof. Dr. Yücel SAYGIN ....

DATE OF APPROVAL: ....

to the ones beyond the time

Acknowledgements

I am deeply grateful to my advisor, Özgür Kıbrıs for his helpful comments and suggestions throughout this work. I am also appreciative to my thesis jury members, Mehmet Baç and Yücel Saygın for their helpful comments about my thesis. I also thank Saadet Saraç, İrfan Sağdınç and Fevzi Kürşad Enç for providing me the information about SUDO's room assignment procedure.

AN ANALYSIS OF A REAL-LIFE ALLOCATION PROBLEM

Mahmut Kemal ÖZBEK
Economics, M.A. Thesis, 2007
Supervisor: Özgür KIBRIS

Abstract

We consider a real-life problem faced by the Sabancı University Dormitory Office (SUDO). Every year SUDO (i) allocates the dormitory beds among applicants and then (ii) determines the roommates that will share each room. For the allocation part, we examine the allocation rule that is currently used and we show that it does not satisfy Pareto efficiency, strategy-proofness and justified no envy. To eliminate these shortcomings, we introduce a modified version of the well-known serial dictatorship rule. We then analyze the roommate assignment rule that is currently used by SUDO. We determine that this rule also has serious shortcomings such as producing unstable and Pareto inefficient matchings. We then modify the rule to eliminate these failures. Moreover, we introduce a new kind of roommate problem in which each agent has three roommates. We then obtain some conditions which guarantee the existence of a stable matching for this kind of roommate problem.

Keywords: Allocation problem, justified envy, roommate problem, stability

BİR GERÇEK HAYAT DAĞITIM PROBLEMİNİN İNCELEMESİ

Mahmut Kemal ÖZBEK

Ekonomi, Yüksek Lisans Tezi, 2007

Tez Danışmanı: Özgür KIBRIS

Özet

Sabancı Üniversitesi Yurt Ofisi'nin (SÜYO) karşılaştığı bir gerçek hayat problemini inceledik. Her sene SÜYO (i) yurt yataklarını başvuranlar arasında dağıtıyor ve (ii) her bir odayı paylaşacak oda arkadaşlarını belirliyor. Dağıtım kısmı için kullanılan kuralı inceledik ve gösterdik ki bu kural Pareto verimlilik, strateji korunumluluk ve mazur gösterilemez öykünüm özelliklerini sağlamıyor. Kuralın bu eksikliklerini gidermek için, çok iyi bilinen dizisel diktatörlük kuralını değiştirerek uyguladık. Daha sonra, SÜYO tarafından oda arkadaşı kısmı için kullanılan kuralı inceledik ve bu kuralın ise kararsız ve Pareto verimsiz eşleşmeler ürettiğini tespit ettik. Bu eksiklikleri yok etmek için kuralda değişiklikler yaptık. Bunlardan başka, her bir ajanın üç tane oda arkadaşı olduğu yeni bir tür oda arkadaşı problemi ortaya koyduk. Ayrıca, bu yeni tür oda arkadaşı problemi için kararlı eşleşmelerin varlığını sağlayacak çeşitli koşullar öne sürdük.

Anahtar Sözcükler: Dağıtım problemi, mazur gösterilebilir öykünüm, oda arkadaşı problemi, kararlılık

1 Introduction

In this work, we examine the following real-life problem. Each year, Sabancı University Dormitory Office (hereafter, SUDO) allocates dormitory rooms among students according to their assigned priorities and preferences. SUDO uses a procedure for this allocation problem. This allocation procedure has three stages. The first stage is the selection of the students that will get a bed and determination of which type of bed they will get. The second stage is the formation of the roommates among the students who got a bed of the same type in the first stage. And the third stage is the assignment of the students to the rooms by using the roommates information from the second stage.

However, when we analyze an outcome of this procedure for some problem, we observe that it can be unfair and inefficient. In this work, our objective is to propose an alternative procedure which solves the unfair issues and improves the inefficient results of the SUDO procedure.

Our paper contributes to two strands of literature. First, the literature on allocation theory. Second, the literature on matching theory, built on a seminal paper by Gale and Shapley (1962). Our contribution is three-fold. First, we present an application of theoretical results in these areas. Second, we extend existing models and results in allocation theory to allow constraints due to gender differences. Third, we widen current models and results in matching theory by allowing number of roommates to be more than two.

There are two types of rooms in Sabancı University's (hereafter, SU) dormitories: the rooms with two beds and the rooms with four beds (which also differ with respect to cost and space). Every student has preferences over these different kinds of rooms. Besides this, the students want to stay in a room with their friends. Thus the students have also preferences over potential roommates.

Every year, the number of students who want a room exceeds the total number of beds at dormitories (see Table 1). Therefore, a subset of the students has to be selected. For this purpose, each student is ordered with respect to previously defined priorities and each one of them is asked to declare his or her preferred room type: 2-bedroom or 4-bedroom. Prior to this, the students are already in two separate groups with respect to their gender. These gender groups are formed because SUDO forbids the students of opposite sexes being assigned to the same room. Then SUDO uses a student selection rule that determines which students will get a bed and which type of bed those selected students will get based on these assigned priorities, submitted preferences and gender information.

Analyzing the SUDO student selection rule, we see that it can produce "unfair" solutions. Precisely, there may be a student whose ranking is high but who does not get a bed. At the same time, there may be another student of the same gender whose ranking is lower but who gets a bed. This situation is called *same gender justified envy* (hereafter, sg-justified envy). In a closely related problem ("school choice problem") Abdulkadiroğlu and Sönmez (2003) defined this situation, where the students are not necessarily having the same gender, as justified envy. In a solution not having sg-justified envy, there should be no unmatched student-room pair (i, r) where student i prefers room r to not being assigned a bed and i has higher priority than some other student j of the same gender who is assigned a bed in room r. This problem arises since the SUDO rule only considers students' first choice of room type.

In the literature, this first stage of the problem is widely discussed for allocation of dormitory rooms (or on-campus housing facilities) to students (Hylland and Zeckhauser (1979)). The following rule, which is known as the *serial dictatorship*, is almost exclusively used in real-life applications of these problems (Abdulkadiroğlu and Sönmez (1998, 1999)): First order students according to some priority. Then assign the first student his first choice, the next student his top choice among the remaining slots, and so on. This rule is not only *Pareto efficient*, but also *strategy-proof* (that is, it can not be manipulated by students who misrepresent their preferences), and it can accommodate any hierarchy of seniorities. It also eliminates *sg-justified envy.*

A major concern of the institution that implements a dormitory room assignment procedure might be to represent a certain balance between students of different genders. For the school choice problem, Abdulkadiroğlu and Sönmez (2003) discuss a similar issue for racial concerns and define the situation where there are quotas for different types of students as *controlled choice*. We call this version of the room assignment problem as a *controlled student selection problem*. An important advantage of the serial dictatorship rule is that it can be easily modified to accommodate controlled student selection constraints by imposing gender quotas. Furthermore, the modified rule is still *strategy-proof* and *constrained efficient*. Also, it still eliminates *sg-justified envy*.

After selecting the students and determining which type of beds they will get, SUDO uses an algorithm to assign each type of selected students to their actual rooms. While doing this, SUDO considers students' priority orders and previously defined room orders. In addition to these criteria, SUDO also considers the students' desire of being assigned to a room with their friends. For this purpose, every student is asked to declare the list of his or her desired roommates. The outcome of this algorithm consists of separate groups of students. We call such an outcome a matching.

The problem with the SUDO roommate algorithm is that it can produce "unstable matchings". A group of 4 (or 2) students block a matching if as roommates they all prefer the group members to their existing roommates. A matching is stable if it can not be blocked[1].

Another shortcoming of the SUDO algorithm is that its matching can be Pareto dominated. In other words, a re-formation of the groups can be beneficial for all students.

In the literature, the problem of forming groups among 2-bedroom type male students or among 2-bedroom type female students is known as the roommate problem (Gale and Shapley (1962)). A roommate problem involves a set of even cardinality n, each member of which ranks all the others in order of preference. Therefore, a stable matching is a partition of this single set into $n/2$ pairs so that no two unmatched members both prefer each other to their partners under the matching. However, the roommate problem need not to have a stable solution.

[1] A central issue in the matching theory literature is to find a stable matching. However, many problems do not have a stable solution. See Alkan (1986), Gale and Shapley (1962), Roth and Sotomayor (1990) for cases where stable matchings fail to exist.

Some further exploration of the roommate problem are considered by Granot (1984), Gusfield (1988), and Irving (1986). Irving (1986) observes, among other things, that the task of finding stable matchings in the roommate problem is a generalization of the same task in the marriage problem[2]. He proposes an efficient algorithm which detects whether a roommate problem has a stable matching and finds one if there is any. Moreover, Tan (1991) proposes a necessary and sufficient condition which guarantees a stable matching for a roommate problem when the agents possess strict preferences.

Chung (2000) points to the restriction on agents' preferences in a marriage problem which makes the problem a special case of the roommate problem. He then asks whether there are other restrictions which provide the roommate problem to have a stable solution. He proposes a sufficient condition called "no odd rings" for a roommate problem to have a stable solution even when the preferences are not strict. Besides, he gives economically more intuitive conditions which implies the no odd rings condition such as agents having "dichotomous preferences". He also shows that the Roth-Vande Vate (1990) process (which is originally proposed for the marriage problem to find a stable matching by starting from a random matching and satisfying each blocking pair whenever there is one) can be used for the roommate problem to find a stable matching whenever the no odd rings condition holds.

However, the problem of forming groups among 4-bedroom type male students or among 4-bedroom type female students is different from the classical roommate problem defined above. Now the problem involves a set of cardinality n which is divisible by 4 and a solution to this problem is a partition of this single set into $n/4$ separate subsets. Therefore, every subset consists of 4 students and these students are now called roommates. We again call an outcome of this problem a matching. Here again the central issue is to find a stable matching for this problem. The results for the classical roommate problem can be adopted to this kind of problem while searching for a stable matching.

The remainder of the paper is organized as follows. In Section 2, we define the student

[2] A marriage problem is that of matching n men and n women, each of whom has ranked the members of the opposite sex in order of preference (Gale and Shapley (1962)).

selection problem. In Subsection 2.1 we analyze SUDO's rule. In Subsection 2.2 we modify the SUDO rule by using the results in the literature. In Subsection 2.3 we analyze the problem under quota restrictions. In Section 3, we define the roommate problem for 2-bedroom type and 4-bedroom type students. In Subsection 3.1 we examine SUDO's rule for 2-bedroom type students and evaluate the rule by the results in the literature. In Subsection 3.2 we examine SUDO's rule for 4-bedroom type students and propose some results considering the existence of a stable matching. Section 4 concludes with a list of open questions.

2 Student Selection Problem

In the student selection problem, there are a number of students, each of whom want to be assigned a bed at one of a number of dormitories. Each dormitory has a maximum number of beds and the number of students exceeds the total number of beds in dormitories. In SU, there are two types of dormitories which differ by their rooms' bed capacities. One type of dormitory (hereafter, type 2 dormitory) has rooms all of which have 2 beds (hereafter, type2 room) and the other type (hereafter, type 4 dormitory) has rooms all of which have 4 beds (hereafter, type4 room). These different types of rooms also differ with respect to cost and space.

Each student has strict preferences over different types of rooms. Despite the fact that the rooms of the same type may differ by many features (such as being at different dormitories), in this stage of the problem each student is assumed to be indifferent between the rooms of the same type. The reason behind this assumption is that the students are not assigned their specific rooms in this stage; only a subset of the students is selected and which type of room these selected students will get is determined.

A strict ordering is constructed according to previously defined priorities by SUDO. Here, priorities do not represent the SUDO's preferences but they are imposed by the SUDO's rigid rules. For example, a senior student is given priority for the rooms. Similarly, a student who has a dormitory scholarship is given priority. These priorities will be explained in detail in the next subsection. Since every student is treated equally except their priorities, this unique strict ordering of students is used by all the rooms during the selection process.

Formally, the **student selection problem** is defined as follows: The finite set of students who want a bed at one of the dormitories is N. The set of all **male students** is M, and the set of all **female students** is F. Hence $N = M \cup F$. For simplicity, we treat the union of type 2 dormitories as one dormitory and denote it as D^2 and similarly denote the union of type 4 dormitories as D^4. Therefore, dormitory D^2 is the set of type2 rooms and dormitory D^4 is the set of type4 rooms. A typical **room** of D^2 is denoted by r^2 and a typical room of D^4 is denoted by r^4.

There are three types of beds: First, a bed b is a b^2 type if it is in D^2. Second, a bed is a b^4 type if it is in D^4. And third, a bed is a $\varnothing$ type if it is neither in D^2 nor in D^4. The set of these types is denoted by X, that is $X = \{b^2, b^4, \varnothing\}$. There is an excess demand for beds in SU. Therefore, let $D = D^2 \cup D^4$ as the set of all rooms and $B = \{b \in r | \forall r \in D\}$ as the set of all beds, then $|B| = 2|D^2| + 4|D^4| < |N|$. An indicator function T, which is defined as $T : B \to \{b^2, b^4\}$, gives the type of a bed in B.

There is an asymmetric and negatively transitive binary relation on N denoted by θ which is determined from previously defined priorities by SUDO. We call this relation **priority ordering**. For the negation of θ, we will use $\tilde{\theta}$. Asymmetry requires that for each i and j in N, $i\theta j$ implies $j\tilde{\theta}i$ and negative transitivity requires that for each $i, j, k \in N$, $i\tilde{\theta}j$ and $j\tilde{\theta}k$ implies $i\tilde{\theta}k$. Also θ is assumed to be weakly connected. Weakly connectedness requires that for each $i, j \in N$, either $i = j$ or $i\theta j$ or $j\theta i$. Each student i's order in the priority ordering is denoted by θ^i. For example, for the first student $i \in N$, $\theta^i = 1$ and for the last student $j \in N$, $\theta^j = |N|$. A **gender function** g defined as $g : N \to \{m, f\}$ indicates the gender of a student in such a way: If a student i is male then, g maps i to m, but if i is female it maps i to f.

Each student i is assumed to have an asymmetric, negatively transitive and weakly connected **preference relation** P_i on X. Hence, i's preferences might be of the form

$b^2 P_i b^4 P_i \varnothing$ indicating that i's first choice, which is denoted by P_i^1, is to be assigned a b^2 type bed, his second choice, which is denoted by P_i^2, is to be assigned a b^4 type bed, and his third choice, which is denoted by P_i^3, is to be assigned an $\varnothing$ type bed. Note that, by the definition of N, there can not be any student $i \in N$ where $P_i^1 = \varnothing$. We will use $\tilde{P}_i$ for the negation of P_i.

The set of all preference relations on X is $\mathcal{P}$. A vector consisting of every student's preference relations is called a **preference profile** and is denoted by $P = (P_1, ..., P_{|N|})$. P_{-i} denotes a vector of preference relations of students other than i. Hence, $P = (P_i, P_{-i})$ is a preference profile. Similarly, for any coalition $C \subseteq N$, $P_C = (P_i)_{i \in C}$ and $P_{-C} = (P_i)_{i \in N \setminus C}$. The set of all preference profiles is denoted by $\mathcal{P}^N$.

The student selection problem is a vector consisting of the set of students, the set of rooms, the priority ordering, and a preference profile: (N, D, θ, P). However, since in our model only P can be different between any two different student selection problems, with abuse of notation we will use P also for a **student selection problem**. By the same reasoning, the set of all student selection problems is denoted by $\mathcal{P}^N$.

The outcome of the student selection problem is an assignment of students to the bed types and we call each such outcome a **selection**[3]. Therefore, a selection σ is a vector in X^N. The student i's assigned bed type under σ is σ_i.

Every selection decomposes N into three disjoint sets as follows: N^2 is the set of students who will get a b^2 type bed (that is, $N^2 = \{i \in N | \sigma_i = b^2\}$), N^4 is the set of students who will get a b^4 type bed (that is, $N^4 = \{i \in N | \sigma_i = b^4\}$), and $N^{\varnothing}$ is the set of students who will get neither a b^2 type bed nor a b^4 type bed (that is, $N^{\varnothing} = \{i \in N | \sigma_i = \varnothing\}$). The union of the sets N^2 and N^4 is denoted by N^s and it refers to **the set of selected students** determined by this selection. These two sets, N^2 and N^4 are also decomposed into two separate sets due to gender respectively. These four sets are as follows: $M^2 = \{i \in N^2 | g(i) = m\}$, $F^2 = \{i \in N^2 | g(i) = f\}$, $M^4 = \{i \in N^4 | g(i) = m\}$ and $F^4 = \{i \in N^4 | g(i) = f\}$.

[3] Indeed, a selection is a matching between students and bed types where each bed type can be matched to more than one student, but each student can only be matched to one bed type.

A selection σ is a **student selection** for a student selection problem when the following *SUDO conditions* are satisfied:

1. $|M^2| + |F^2| \leq 2|D^2|$ and when this is an equality, both $|M^2|$ and $|F^2|$ are divisible by 2

2. $|M^4| + |F^4| \leq 4|D^4|$ and when this is an equality, both $|M^4|$ and $|F^4|$ are divisible by 4

The set of all student selections for a student selection problem is denoted by Σ. A student selection $\sigma \in \Sigma$ is *Pareto efficient* if there does not exist any $\sigma' \in \Sigma$ such that for each $i \in N$, $\sigma_i \tilde{P}_i \sigma'_i$ and there exits at least one $i \in N$ where $\sigma'_i P_i \sigma_i$.

A *student selection rule* (hereafter, SSR) $\mathcal{S}$ is a systematic procedure that produces a student selection for each student selection problem. That is, $\mathcal{S} : \mathcal{P}^N \to \Sigma$. An SSR $\mathcal{S}$ is Pareto efficient if for each $P \in \mathcal{P}^N$, $\mathcal{S}(P)$ is Pareto efficient. An SSR $\mathcal{S}$ is *strategy-proof* if for each $i \in N$ and for each $P \in \mathcal{P}^N$, there does not exist any $P'_i \in \mathcal{P}$ such that $\mathcal{S}_i(P'_i, P_{-i}) P_i \mathcal{S}_i(P_i, P_{-i})$. An SSR $\mathcal{S}$ is *coalitional strategy-proof* if for any $C \subseteq N$, for any $P = (P_C, P_{-C}) \in \mathcal{P}^N$ and for any $P' = (P'_C, P_{-C}) \in \mathcal{P}^N$, there exists an $i \in C$ such that $\mathcal{S}_i(P) P_i \mathcal{S}_i(P')$. An SSR $\mathcal{S}$ eliminates *same gender justified envy* (hereafter, sg-justified envy) if for each $P \in \mathcal{P}^N$ and for each $i \in N$, $\{j \in N | [\mathcal{S}_j(P) P_i \mathcal{S}_i(P)] \wedge [\theta^j > \theta^i] \wedge [g(j) = g(i)]\} = \emptyset$. An SSR $\mathcal{S}$ eliminates *opposite gender justified envy* (hereafter, og-justified envy) if for each $P \in \mathcal{P}^N$ and for each $i \in N$, $\{j \in N | [\mathcal{S}_j(P) P_i \mathcal{S}_i(P)] \wedge [\theta^j > \theta^i] \wedge [g(j) \neq g(i)]\} = \emptyset$. An SSR eliminates *justified envy* if it eliminates both sg-justified envy and og-justified envy.

Since it is not possible to assign each student his top choice, a central issue in the student selection problem is the design of a "good" rule. We now first describe and analyze the student selection rule used by SUDO.

2.1 SUDO Student Selection Rule (SUDO-SSR)

Prior to 2005, SUDO officers manually selected the students who would get a bed. After 2005, to be more objective and to speed up the process, SUDO started to use the following mechanism for the first stage of the room assignment procedure.

SUDO-SSR works as follows:

1. Each student submits a choice of room type.

2. A priority ordering is determined according to the following criteria:

   - First priority: Having a dormitory scholarship
   - Second priority: Coming from out of the city and being a senior student
   - Third priority: Coming from out of the city and being a junior student
   - Fourth priority: Coming from out of the city
   - Fifth priority: Being a senior student
   - Sixth priority: Being a junior student
   - Seventh priority: Coming from the European part of the city
   - Eight priority: Coming from the Anatolian part of the city (far)
   - Ninth priority: Coming from the Anatolian part of the city (nearby)

3. Students in the same priority group are ordered based on the following hierarchy:

   - First priority: University entrance ranking
   - Second priority: Birth date (being young is better)
   - Third priority: University ID number (having a smaller number is better)

Item 2 and 3 determine a unique θ for the students.

4. The final phase is the selection of students based on priorities, preferences and gender:

Associate a counter to each dormitory as follows: c_2 and c_4 keep track of how many beds are still available in D^2 and D^4 respectively. Initially $c_2 = 2|D^2|$, and $c_4 = 4|D^4|$. Also, put four gender-bed counters as follows: c^2_m and c^4_m count respectively how many b^2 and b^4 type of beds will be assigned to male students. c^2_f and c^4_f count respectively how many b^2 and b^4 type of beds will be assigned to female students. Initially, all gender-bed counters are equal to zero.

Step 1: Start with student i in N with $\theta^i = 1$ in the priority ordering and assign i to the corresponding bed type according to i's submitted choice. Depending on i's choice, the associated dormitory counter is reduced by one. Depending on i's choice and gender, the associated gender-bed counter is incremented by one. The other counter stays put.

In general at

Step k: Consider student i in N with $\theta^i = k$.

Case 1 [i's submitted choice is b^2 and $0 \leq c_2 \leq 1$ and $c^2_{g(i)}$ is divisible by 2]: Assign i to $\varnothing$. All the counters remain the same.

Case 2 [i's submitted choice is b^4 and $0 \leq c_4 \leq 3$ and $c^4_{g(i)}$ is divisible by 4]: Assign i to $\varnothing$. All the counters remain the same.

Case 3 [Otherwise]: Assign i to the corresponding bed type according to i's choice. Depending on i's choice, the associated dormitory counter is reduced one. Depending on i's choice and gender, the associated gender-bed counter is incremented by one. The other counter stays put.

The algorithm terminates when $c_2 = c_4 = 0$. All the remaining students are assigned to $\varnothing$.

Note that, the SUDO-SSR algorithm only uses the top bed type choice of the students.

The major difficulty with the SUDO-SSR is that it may not eliminate sg-justified envy as the following example suggests:

Example 1 *There are* 8 *students of the same gender,* $N = \{i_1, ..., i_8\}$ *and there are two rooms* r^2 *and* r^4 *consisting of* 2 *and* 4 *beds respectively. The priority ordering for each* $i_k \in N$ *is such that* $\theta^{i_k} = k$*. The preferences are as follows:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$	$\mathbf{i}_7$	$\mathbf{i}_8$
b^2	b^2	b^2	b^4	b^4	b^4	b^4	b^2
b^4	b^4	b^4	b^2	b^2	b^2	b^2	b^4
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

For these priorities and preferences, SUDO-SSR produces the student selection σ which assigns students i_1, i_2 to b^2, students i_4,i_5, i_6, i_7 to b^4, and students i_3, i_8 to $\varnothing$. However, $\theta^{i_3} < \theta^i$ for any $i \in \{j \in N|\sigma_j = b^4\}$ and $b^4 P_{i_3} \varnothing$.

Here, after assigning i_1, i_2 to b^2, SUDO-SSR considers i_3's first choice. But since r^2 is now full, it can not assign i_3 to b^2. However, instead of considering i_3's second choice, SUDO-SSR directly assigns i_3 to $\varnothing$ which is i_3's third choice.

Since there is a threat of not getting a bed in SU dormitories even for the high ranked students when they reveal their true preferences, students may misrepresent unilaterally their preferences to benefit from this selection mechanism. Because of this, SUDO-SSR is not strategy-proof. In the above example, student i_3 is assigned to $P_{i_3}^3 = \varnothing$. He may instead declare his preference relation as $b^4 P_{i_3} b^2 P_{i_3} \varnothing$ and will be assigned to $P_{i_3}^2 = b^4$ instead of $P_{i_3}^3 = \varnothing$.

Another difficulty with the SUDO-SSR concerns efficiency. If students submit their true preferences, then the outcome of the SUDO-SSR is Pareto efficient. But since many students are likely to misrepresent their preferences, its outcome is unlikely to be Pareto efficient. The following example illustrates this situation:

Example 2 *There are* 8 *students of the same gender,* $N = \{i_1, ..., i_8\}$ *and there are two rooms* r^2 *and* r^4 *consisting of* 2 *and* 4 *beds respectively. The priority ordering for each* $i_k \in N$ *is such that* $\theta^{i_k} = k$. *The preferences are as follows:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$	$\mathbf{i}_7$	$\mathbf{i}_8$
b^2	b^4	b^2	b^4	b^4	b^4	b^4	b^2
b^4	b^2	b^4	b^2	b^2	b^2	b^2	b^4
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

For these priorities and preferences, SUDO-SSR produces the student selection σ which assigns students i_1, i_3 to b^2, students i_2, i_4,i_5, i_6 to b^4, and students i_7, i_8 to $\varnothing$. But i_3 may believe that i_2's preferences is such that $b^2 P_{i_2} b^4 P_{i_2} \varnothing$. If this was the case, then SUDO-SSR would produce the student selection σ' which assigns students i_1, i_2 to b^2, students i_4,$i_5, i_{6,} i_7$ to b^4, students i_3, i_8 to $\varnothing$. By the threat of not getting a bed in SU dormitories, i_3 may change his true preferences in such a way: $b^4 P'_{i_3} b^2 P'_{i_3} \varnothing$.

For these preferences, SUDO-SSR will produce the student selection σ'' which assigns students i_1, i_8 to b^2, students i_2, i_3, i_4,i_5 to b^4, students i_6, i_7 to $\varnothing$. However, at the same time i_6 may believe that i_3's preferences is such that $b^4 P_{i_3} b^2 P_{i_3} \varnothing$ (indeed it is a true belief when i_3 misrepresents as above). Therefore, by the threat of not getting a bed, i_6 may change his true preferences in such a way: $b^2 P_{i_6} b^4 P_{i_6} \varnothing$.

For these preferences, SUDO-SSR produces the student selection σ''' which assigns students i_1, i_6 to b^2, students i_2, i_3, i_4,i_5 to b^4,and students i_7, i_8 to $\varnothing$. However, now this situation occurs: $\sigma'''_{i_6} P_{i_3} \sigma'''_{i_3}$ and $\sigma'''_{i_3} P_{i_6} \sigma'''_{i_6}$.

2.2 Gender Sensitive Serial Dictatorship Rule (GS-SDR)

In the previous section, we see that SUDO's rule has serious shortcomings. The fact that SUDO-SSR does not use full preference information causes these failures. If we consider the students' full preferences, then these problems may disappear. For this purpose, we could use a modified *Step k* of the SUDO-SSR as follows:

Step k: Consider the student i in N with $\theta^i = k$ and consider P_i^1.

Case 1 [$P_i^1 = b^2$ and $0 \leq c_2 \leq 1$ and $c^2_{g(i)}$ is divisible by 2]: Consider P_i^2.

Case 2 [$P_i^1 = b^4$ and $0 \leq c_4 \leq 3$ and $c^4_{g(i)}$ is divisible by 4]: Consider P_i^2.

Otherwise, assign i to the corresponding bed type according to P_i^1. Depending on P_i^1, the associated dormitory counter is reduced one. Depending on P_i^1 and $g(i)$, the associated gender-bed counter is incremented by one. The other counter stays put.

Case 3 [$P_i^2 = b^2$ and $0 \leq c_2 \leq 1$ and $c^2_{g(i)}$ is divisible by 2] : Assign i to $\varnothing$. All the counters remain the same.

Case 4 [$P_i^2 = b^4$ and $0 \leq c_4 \leq 3$ and $c^4_{g(i)}$ is divisible by 4]: Assign i to $\varnothing$. All the counters remain the same.

Otherwise, assign i to the corresponding bed type according to P_i^2. Depending on P_i^2, the associated dormitory counter is reduced one. Depending on P_i^2 and $g(i)$, the associated gender-bed counter is incremented by one. The other counter stays put.

The algorithm terminates when $c_2 = c_4 = 0$. All the remaining students are assigned to $\varnothing$.

GS-SDR annihilates the failures of SUDO-SSR as the following propositions state:

Proposition 3 *For every student selection problem P, GS-SDR eliminates sg-justified envy.*

Proof. Suppose that there exits sg-justified envy in an outcome σ of GS-SDR for a student selection problem P. Then, there must be a student $i \in N$ where $\{j \in N | [\sigma_j P_i \sigma_i] \wedge [\theta^j > \theta^i] \wedge [g(j) = g(i)]\} \neq \emptyset$. Consider a student j in this set. Since $\forall k \in N$, $P_k^1 \neq \varnothing$ and $\sigma_k = P_k^3 \Rightarrow P_k^3 = \varnothing$, and since for i and j, $\sigma_j P_i \sigma_i$, then it must be the case that $\sigma_j \neq \varnothing$. Since $\theta^j > \theta^i$, at step θ^i, it must be the case that either $c_2 \geq 1$ or $c_4 \geq 1$ according to σ_j. But then since $g(j) = g(i)$, i must be assigned to σ_j at step θ^i. This is the required contradiction. ■

Remark 4 *GS-SDR may not eliminate og-justified envy in some situations. However, this is caused by the SUDO's requirement which states that students with different genders can not be assigned to the same room. The following example illustrates this situation:*

Example 5 *There are* 8 *students,* $N = \{i_1, ..., i_8\}$ *and there are two rooms* r^2 *and* r^4 *consisting of* 2 *and* 4 *beds respectively. The priority ordering for each* $i_k \in N$ *is such that* $\theta^{i_k} = k$*.* i_1, i_4, i_8 *are female and the others are male students. The preferences are as follows:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$	$\mathbf{i}_7$	$\mathbf{i}_8$
b^2	b^4	b^2	b^4	b^4	b^4	b^4	b^2
b^4	b^2	b^4	b^2	b^2	b^2	b^2	b^4
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

For these priorities, preferences and gender information, GS-SDR produces the student selection σ which assigns students i_1, i_4 to b^2, students i_2, i_3, i_5, i_6 to b^4, and students i_7, i_8 to $\varnothing$. Here, $\sigma_{i_3} = b^4$ and $\sigma_{i_4} = b^2$. However, $\theta^{i_3} < \theta^{i_4}$ and $\sigma_{i_4} P_{i_3} \sigma_{i_3}$. Hence, there is og-justified envy in this selection.

In fact, there can not be any rule which eliminates og-justified envy when GS-SDR can not do so.

Next we analyze the strategic properties of GS-SDR.

Proposition 6 *GS-SDR is strategy-proof.*

Proof. Consider a student selection problem P and a student $i \in N$. We want to show that revealing his true preferences P_i is at least as good for i as declaring any other preferences $P'_i \in \mathcal{P}$. Construct a new problem P' by letting $P' = (P'_i, P_{-i})$. Since the priority order does not change, the students are considered at the same steps in both of these problems. Moreover, any student j with $\theta^i > \theta^j$ is assigned to the same bed type in both P and P' since j has the same preferences in both problems, that is $P'_j = P_j$.

At step θ^i, if student i is assigned to P_i^1, then he will not have an incentive to misrepresent his preferences. Therefore, assume that in P he is assigned to P_i^k where $k \neq 1$. Since GS-SDR first considers P_i^1, at step θ^i, it must be the case that P_i^1 bed type is not available for i. If in P, i is assigned to P_i^3, then by the same reason P_i^2 bed type is also not available for

i. However, changing the order of bed types in his preferences will not change this situation for him. Therefore, he can not get a much preferred bed type in P'. ■

GS-SDR is coalitional strategy-proof since SUDO does not allow the students to exchange their rooms. However, if room exchange is permitted, then GS-SDR will not be coalitional strategy-proof as the following example suggests:

Example 7 *There are* 8 *students of the same gender,* $N = \{i_1, ..., i_8\}$ *and there are two rooms* r^2 *and* r^4 *consisting of* 2 *and* 4 *beds respectively. The priority ordering for each* $i_k \in N$ *is such that* $\theta^{i_k} = k$. *The preferences are as follows:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$	$\mathbf{i}_7$	$\mathbf{i}_8$
b^2	b^4	b^2	b^4	b^4	b^4	b^4	b^2
b^4	b^2	$\varnothing$	b^2	b^2	b^2	b^2	b^4
$\varnothing$	$\varnothing$	b^4	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

For these priorities and preferences, GS-SDR produces the student selection σ which assigns students i_1, i_3 to b^2, students i_2, i_4, i_5, i_6 to b^4, and students i_7, i_8 to $\varnothing$. There, i_2 and i_7 may form a coalition and misrepresent their preferences as follows: $b^2 P_{i_2} b^4 P_{i_2} \varnothing$ and $b^2 P_{i_7} b^4 P_{i_7} \varnothing$. But then, GS-SDR produces the student selection σ' which assigns students i_1, i_2 to b^2, students i_4, i_5, i_6, i_7 to b^4, and students i_3, i_8 to $\varnothing$. After they are assigned to their actual rooms, i_2 and i_7 can exchange their rooms.

We had noted that the SUDO-SSR is not efficient. Next, we will explore efficiency properties of GS-SDR.

Proposition 8 *GS-SDR is Pareto efficient.*

The intuition for the Pareto efficiency of the GS-SDR is very simple. By the rule GS-SDR, the first student in the priority ordering gets his best bed type. Therefore, he can not be made better-off. Then the second student gets his best type among the remaining ones. Therefore, he can not also be made better-off unless the first one is made worse-off. Continuing in this way, we will reach the result that no one can be made better-off without hurting someone. But different from this approach, we prove the proposition in the Appendix by contradiction.

2.3 Controlled Student Selection

Controlled student selection attempts to select students to determine which ones will get a bed while maintaining the gender balance at dormitories. Prior to 2006, controlled selection constraints were implemented by imposing gender quotas at SU dormitories. SUDO was determining some rooms available only for the female students and the others available only for the male ones. This type of controlled selection constraint is perfectly rigid. For such a situation, there is no need to modify serial dictatorship rule. For each gender, one can separately implement the rule in order to allocate the beds that are reserved exclusively for that gender.

However, controlled selection constraints may be flexible. For example, consider 100 beds and assume that SUDO determines the average enrollment rates of male students versus female ones as 45%, 55% respectively, and allows these rates to go above or below up to 5 percent points. Gender quotas for this student selection problem are 50 for male students, and 60 for female ones. Serial dictatorship can be easily modified to accommodate controlled selection constraints by imposing type-specific quotas.

2.3.1 Serial Dictatorship Rule with Type-Specific Quotas over Rooms (SDR-TSQR)

If these type-specific quotas are imposed separately for each type of rooms, then the following rule could be used: Consider D^2 with q_2 rooms and which has quotas of q_2^m, q_2^f for male, female students respectively. Clearly $q_2 \geq q_2^m$, $q_2 \geq q_2^f$ and $q_2^m + q_2^f \geq q_2$. In D^2:

- $q_2 - q_2^m$ rooms are reserved exclusively for male students,

- $q_2 - q_2^f$ rooms are reserved exclusively for female students,
- and the remaining $q_2^m + q_2^f - q_2$ rooms are reserved for either type of students.

Similarly consider D^4 with q_4 rooms and which has quotas of q_4^m, q_4^f for male, female students respectively. Clearly $q_4 \geq q_4^m$, $q_4 \geq q_4^f$ and $q_4^m + q_4^f \geq q_4$. In D^4:

- $q_4 - q_4^m$ rooms are reserved exclusively for male students,
- $q_4 - q_4^f$ rooms are reserved exclusively for female students,
- and the remaining $q_4^m + q_4^f - q_4$ rooms are reserved for either type of students.

So it is as if there are three different dormitories d^m, d^f, and d^b where

- dormitory d^m has $(q_2 - q_2^m)$ type2 and $(q_4 - q_4^m)$ type4 rooms and student priorities are obtained from the original priorities by removing the female students and making them unacceptable at dormitory d^m. For this smaller problem, we could use serial dictatorship rule and determine a student selection.
- dormitory d^f has $(q_2 - q_2^f)$ type2 and $(q_4 - q_4^f)$ type4 rooms and student priorities are obtained from the original priorities by removing the male students and making them unacceptable at dormitory d^f. For this smaller problem, we could use serial dictatorship rule and determine a student selection.
- dormitory d^b has $(q_2^m + q_2^f - q_2)$ type2 and $(q_4^m + q_4^f - q_4)$ type4 rooms and those students who are not selected in above problems are acceptable at dormitory d^b. Their priorities are obtained from the original priorities by removing the students who are selected already in the above problems. For this smaller problem, we could use GS-SDR and determine a student selection.

Corollary 9 *SDR-TSQR is strategy-proof and it eliminates sg-justified envy.*

Proof. Since both SDR and GS-SDR are strategy-proof rules, then every outcome of SDR-TSQR is strategy-proof. And since both SDR and GS-SDR eliminate sg-justified envy, then SDR-TSQR also eliminates sg-justified envy. ■

However, there can be efficiency losses in the outcome of SDR-TSQR due to the controlled selection constraints. The following example illustrates this point:

Example 10 *There are* 8 *students,* $N = \{i_1, ..., i_8\}$ *and there are three rooms* r_1^2 *and* r_2^2 *both consisting of* 2 *beds and* r^4 *consisting of* 4 *beds. The priority ordering for each* $i_k \in N$ *is such that* $\theta^{i_k} = k$. *The students* i_1, i_2, i_5, i_6 *are female and the other students are male. The quotas are such that* $q_2^m = 0$ *and* $q_4^f = 0$. *The preferences are as follows:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$	$\mathbf{i}_7$	$\mathbf{i}_8$
b^4	b^4	b^2	b^2	b^4	b^4	b^2	b^2
b^2	b^2	b^4	b^4	b^2	b^2	b^4	b^4
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

Under these quotas and for these priorities, preferences and gender information, SDR-TSQR produces a student selection which assigns students i_1, i_2, i_5, i_6 to b^2 and assigns students i_3, i_4, i_7, i_8 to b^4. However, students i_1, i_2, i_5, i_6 all prefer b^4 to b^2. At the same time, students i_3, i_4, i_7, i_8 all prefer b^2 to b^4. Therefore, there is an efficiency loss.

A student selection is *constrained efficient* if there is no other selection that satisfies the controlled selection constraints, and which assigns all students to a weakly better bed type and at least one student to a strictly better one. Every outcome of SDR-TSQR is constrained efficient.

Proposition 11 *SDR-TSQR is constrained efficient.*

Proof. Since SDR is Pareto efficient, any student who gets a bed in d^m or d^f cannot be made better off without hurting someone who gets a bed in d^m or in d^f. And also since

GS-SDR is Pareto efficient as SDR, any student who gets a bed in d^b cannot be made better off without hurting someone who gets a bed in one of these three dormitories. Therefore SDR-TSQR is constrained efficient. ■

2.3.2 Serial Dictatorship Rule with Type-Specific Quotas over Beds (SDR-TSQB)

In a more general setting, these type-specific quotas could be imposed for the total number of beds. For example, there could be in total q beds and those beds collectively have quotas of q^m, q^f for male and female students respectively. Clearly $q \geq q^m$, $q \geq q^f$ and $q^m + q^f \geq q$. Then for such a situation a modified version of the GS-SDR can be used as follows:

In addition to dormitory counters c_2, c_4 and gender-bed counters $c_m^2, c_m^4, c_f^2, c_f^4$, associate a counter for each type of students equal to their quota. That is, $c_m = q^m$ and $c_f = q^f$.

Step 1: Start with the student i in N with $\theta^i = 1$ in the priority ordering and assign i to the corresponding bed type according to P_i^1. Depending on P_i^1, the associated dormitory counter is reduced by one. Depending on $g(i)$, the associated type-specific counter is reduced by one. Depending on P_i^1 and $g(i)$, the associated gender-bed counter is incremented by one. The other counters stay put.

In general at

Step k: Consider the student i in N with $\theta^i = k$ and consider P_i^1.

Case 1 [$c_{g(i)} = 0$]: Assign i to $\varnothing$. All the counters remain the same.

Case 2 [$P_i^1 = b^2$ and $0 \leq c_2 \leq 1$ and $c_{g(i)}^2$ is divisible by 2 and $c_{g(i)} \neq 0$]: Consider P_i^2.

Case 3 [$P_i^1 = b^4$ and $0 \leq c_4 \leq 3$ and $c_{g(i)}^4$ is divisible by 4 and $c_{g(i)} \neq 0$]: Consider P_i^2.

Otherwise, assign i to the corresponding bed type according to P_i^1. Depending on P_i^1, the associated dormitory counter is reduced one. Depending on $g(i)$, the associated type-specific counter is reduced by one. Depending on P_i^1 and $g(i)$, the associated gender-bed counter is incremented by one. The other counter stays put.

Case 4 [$P_i^2 = \varnothing$]: Assign i to $\varnothing$. All the counters remain the same.

Case 5 [$P_i^2 = b^2$ and $0 \leq c_2 \leq 1$ and $c^2_{g(i)}$ is divisible by 2 and $c_{g(i)} \neq 0$]: Assign i to $\varnothing$. All the counters remain the same.

Case 6 [$P_i^2 = b^4$ and $0 \leq c_4 \leq 3$ and $c^4_{g(i)}$ is divisible by 4 and $c_{g(i)} \neq 0$]: Assign i to $\varnothing$. All the counters remain the same.

Otherwise, assign i to the corresponding bed type according to P_i^2. Depending on P_i^2, the associated dormitory counter is reduced one. Depending on $g(i)$, the associated type-specific counter is reduced by one. Depending on P_i^2 and $g(i)$, the associated gender-bed counter is incremented by one. The other counter stays put.

The algorithm terminates when $c_2 = c_4 = 0$. All the remaining students are assigned to $\varnothing$.

SDR-TSQB and SDR-TSQR are two closely related rules. First of all, they are both modified versions of SDR. Also, they coincide on a subclass of problems as the following proposition implies. Therefore, some properties of SDR-TSQR can also be acquired by SDR-TSQB.

Proposition 12 *SDR-TSQR produces the same student selection as SDR-TSQB for a controlled student selection problem where the type-specific quotas over rooms are determined by the outcome of SDR-TSQB for the same problem with the type specific quotas over beds.*

Proof. After realizing the student selection for a problem with type-specific quotas over beds by using SDR-TSQB, the problem becomes a controlled student selection problem with perfectly rigid quotas over beds. These perfectly rigid quotas over beds can be transformed to perfectly rigid quotas over rooms for this problem. Then both SDR-TSQR and SDR-TSQB just become the serial dictatorship rule. The only difference with the applications of these rules is that for SDR-TSQR, different types of students are exclusively assigned to bed types, however, for SDR-TSQB they are assigned to bed types in the same process. But since the students priorities and preferences are same in these two applications, then their outcomes will be the same. ■

Now we use this relationship for the following corollary.

Corollary 13 *SDR-TSQB is strategy-proof and constrained efficient. It also eliminates sg-justified envy.*

Proof. As it is stated in the above proposition, by using SDR-TSQR for the corresponding problem with type-specific quotas over rooms, we can have the same student selection for a controlled student selection problem. But we know that this selection is strategy-proof and constrained efficient and it also eliminates sg-justified envy. ■

Remark 14 *However, converse of this proposition is not always true. Explicitly, SDR-TSQB may not produce the same student selection as SDR-TSQR for a problem where the type-specific quotas over beds are determined by the type-specific quotas over rooms. This point can be seen in Example 10.*

3 Roommate Problem

In the previous section, the students who will get a bed and the type of bed they will get were determined. After this determination, there are now four disjoint subsets of selected students which are M^2, F^2 (both have cardinalities divisible by 2) and M^4, F^4 (both have cardinalities divisible by 4). Only the students in one of these subsets are guaranteed a bed and no bed is reserved for more than one student. As a result of this selection, the type 2 dormitory D^2 and type 4 dormitory D^4 are also decomposed into two disjoint subsets respectively as follows: the subset D^2_m (D^2_f) refers to the set of type2 rooms reserved only for the students in M^2 (F^2), and D^4_m (D^4_f) refers to the set of type4 rooms reserved only for the students in M^4 (F^4).

SUDO uses a second algorithm to assign each student in the above subsets to one of the rooms which are exactly reserved for these subsets. To start with, each room is already ordered in its subset by SUDO. This ordering is not based on any criteria. Also, this order information is not known by the students but it is used in the assignment procedure. We associate an "order function" for each of these sets. In addition to room order information and the students' priorities, SUDO also considers the students' desire of being assigned to a room with their friends. For this purpose, every student is asked to declare the list of his or her desired roommates. Therefore, the problem in this section that SUDO deals with is not only an assignment of the rooms, but also a "roommate problem".

Since the situation that the students face is the same for the students in M^2 and for the students in F^2, and similarly it is the same for the students in M^4 and for the students in F^4, in this section we will only consider male students. On the other hand, since a student in M^2 (M^4) can be assigned only a type2 (type4) room and since the number of beds in different types of rooms differs, the number of roommates of the students in M^2 and M^4 differs. Therefore, we will consider the problems for these two sets in separate subsections as follows.

3.1 Roommate Problem for b^2 Type Beds

In a roommate problem for b^2 type beds, there is a set of students denoted by M^2 which has a finite cardinality divisible by 2. Each student i in M^2 is assumed to have a **preference relation** R_i on M^2. We assume that these preference relations are complete, reflexive and transitive. Completeness requires that for any $i, j, k \in M^2$, either jR_ik or kR_ij, reflexivity requires that for any $i, j \in M^2$, jR_ij and transitivity requires that for any $i, j, k, l \in M^2$, jR_ik and kR_il implies jR_il. For the associated **strict preference relation** and **indifference relation**, we will use P_i and I_i respectively.

As before, R_i^1 denotes student i's first choice, R_i^2 denotes his second choice, and so on. For any $i \in M^2$, we let R_i such that for any $j \in M^2$, jR_ii. The set of all preference relations on M^2 is $\mathcal{R}$. A vector consisting of every student's preference relations is called a **preference profile** and is denoted by $R = (R_1, ..., R_{|M^2|})$. The set of all preference profiles is denoted by $\mathcal{R}^{M^2}$.

There is an asymmetric, negatively transitive and weakly connected binary relation on M^2 denoted by θ_{M^2}. In fact, this relation is induced by the priority ordering θ defined on the set of all students N in the previous section. For any student $i \in M^2$, $\theta^i_{M^2}$ is the order of this student and it is defined as follows: $\theta^i_{M^2} = \theta^i - |\{j \in (N \backslash M^2) | \theta^i > \theta^j\}|$.

There is a set of type2 rooms D^2_m where $|D^2_m| = \frac{|M^2|}{2}$. An associated function o^2_m gives the order of each room in D^2_m. That is, $o^2_m : D^2_m \rightarrow \{1, 2, ..., \frac{|M^2|}{2}\}$. We call this function **room ordering**.

The **roommate problem for** b^2 **type beds** (hereafter, b^2-RP) is a vector consisting of the set of students M^2, the priority ordering θ_{M^2}, the set of type2 rooms D^2_m, the room ordering function o^2_m and a preference profile R: $(M^2, \theta_{M^2}, D^2_m, o^2_m, R)$. However, since in our model only R can be different between any two different b^2-RPs, with abuse of notation we will use R also for a b^2-RP. By the same reasoning, the set of all b^2-RPs is denoted by $\mathcal{R}^{M^2}$.

An outcome of a b^2-RP is a partition of M^2 into $|M^2|/2$ disjoint pairs. We call this outcome a **matching** and denote it by μ. In fact, a matching μ is a one-to-one mapping from M^2 onto itself such that for all $\{i, j\} \subset M^2$ where $i \neq j$, $\mu(i) = j$ if and only if $\mu(j) = i$.[4] Each student in such a pair is called the **roommate** of the other student in this pair. The set of all matchings for a b^2-RP is denoted by $\mathcal{M}^2$.

Two students $\{i, j\}$, $i \neq j$ **block** a matching μ if $jP_i\mu(i)$ and $iP_j\mu(j)$. We call such a pair as a **blocking pair**. A central issue for a roommate problem is the existence of a matching in which there are no blocking pairs. If such a matching exists, we say that it is *stable*. A matching $\mu \in \mathcal{M}^2$ is *Pareto efficient* if there does not exist any $\mu' \in \mathcal{M}^2$ such that for each $i \in M^2$, $\mu'(i)R_i\mu(i)$ and there exits at least one $i \in M^2$ where $\mu'(i)P_i\mu(i)$.

A *roommate rule* for b^2-RPs (hereafter, 2-RR) $\mathcal{T}$ is a systematic procedure that produces a matching for each b^2-RP. That is, $\mathcal{T} : \mathcal{R}^{M^2} \rightarrow \mathcal{M}^2$. A 2-RR $\mathcal{T}$ is Pareto efficient if for each $R \in \mathcal{R}^{M^2}$, $\mathcal{T}(R)$ is Pareto efficient.

In the literature, b^2-RP is known as a **roommate problem**. Gale and Shapley (1962)

[4] In the literature, in a matching μ also some students can be matched to himself. That is, $\mu(i) = i$. However, SUDO's objective is to fill all the rooms and so every student must be matched to someone in a b^2-RP.

showed that stable matchings may not exist in a roommate problem with strict preferences. An example is as follows.

Example 15 *Consider $M^2 = \{i, j, k, l\}$ and the following strict preferences:*

i	**j**	**k**	**l**
j	k	i	i
k	i	j	j
l	l	l	k
i	j	k	l

There are no stable matchings for this roommate problem since any matching must pair someone with student l, and that someone will be able to find another person to make a blocking pair. That is, the possible matchings are

$$\mu_1 = \{\{i, j\}, \{k, l\}\}, \mu_2 = \{\{i, l\}, \{j, k\}\}, \mu_3 = \{\{i, k\}, \{j, l\}\}$$

But $\{j, k\}, \{i, k\}$ and $\{i, j\}$ block μ_1, μ_2 and μ_3, respectively.

In the literature, there is also a closely related problem, namely "the marriage problem" (Gale and Shapley (1962)) which is much more fully discussed (see Roth and Sotomayor (1990)). A marriage problem is that of matching n men and n women, each of whom has ranked the members of the opposite sex in order of preference. Indeed, a marriage problem is a special case of the roommate problem. Gale and Shapley (1962) proposed the Gale and Shapley algorithm which produces a stable matching for a marriage problem when the agents' preferences are strict.

Knuth (1976) observed that for the roommate problem with strict preferences, even when there exists a stable matching, Gale and Shapley algorithm may produce an unstable matching for this problem. However, later Irving (1986) introduced an "efficient" algorithm which detects whether a roommate problem with strict preference profile has a stable matching and finds one if there is any. Moreover, Tan (1991) proposed a necessary and sufficient condition which guarantees a stable matching for a roommate problem when the agents possess strict preferences.

Chung (2000) pointed to the restriction on agents' preferences in a marriage problem which makes the problem a special case of the roommate problem. He then asked whether there are other restrictions which provide the roommate problem to have a stable solution. He proposed a sufficient condition called "no odd rings" for a roommate problem to have a stable solution even when the preferences are not strict. Besides, he gave economically more intuitive conditions which implies the no odd rings condition such as agents having "dichotomous preferences" (see Chung (2000) for further survey). He also showed that the Roth-Vande Vate (1990) process (which is originally proposed for the marriage problem to find a stable matching by starting from a random matching and satisfying each blocking pair whenever there is one) can be used for the roommate problem to find a stable matching whenever the no odd rings condition holds.

A preference profile is **dichotomous** if every student classifies every other student into two groups in such a way that he is indifferent among students in each group. Explicitly, for student i, let R_i^1 be the first indifference class and R_i^2 be the second indifference class where $R_i^1 \cup R_i^2 = M^2$. For any $j, k \in R_i^1$, it is the case that jI_ik and for any $l, m \in R_i^2$, lI_im. At the same time, for any $j \in R_i^1$ and for any $l \in R_i^2$, jP_il.

The following proposition is due to Chung (2000).

Proposition 16 *If the preference profile is dichotomous, there exist stable matching for a roommate problem.*

Proof. Different from Chung (2000), we will prove the proposition by using the Roth-Vande Vate (1990) random paths to stability algorithm.

Let μ_1 be an arbitrary matching. Suppose that μ_1 has a blocking pair $\{i_1, i_2\}$. (If no blocking pairs exist, then we are done.) That is, $i_2P_{i_1}\mu_1(i_1)$ and $i_1P_{i_2}\mu_1(i_2)$. Make i_1, i_2 a pair and $\mu_1(i_1), \mu_1(i_2)$ another pair. Let other pairs in μ_1 be the same. Now, we have another matching μ_2. Note that, from now on i_1 or i_2 can never be in another blocking pair since the preferences are dichotomous and so $\nexists j \in M^2$ such that $jP_{i_1}i_2$ and $\nexists k \in M^2$ such that $jP_{i_2}i_1$. Proceed for i_2 and the other matchings that may appear in the same way. Since there is a finite number of students, after a finite step we will get the desired stable matching. ■

To assign the students to the rooms with their friends, SUDO asks each student to declare his roommate list. Therefore, each student classifies every other student into two groups. Specifically for each student i, a student j in his roommate list is in i's top choice indifference class R_i^1 and a student k not in his roommate list is in i's bottom choice indifference class R_i^2. Hence, each student has dichotomous preferences over the set of students. Also, SUDO restricts each student, who will get a b^2 type bed, to declare at most one student in his roommate list. Therefore, each student's top choice indifference class is either singleton or empty.

By the above proposition, we know that there exists a stable matching for a roommate problem with dichotomous preference profile. However, SUDO's roommate algorithm may produce an unstable matching for this problem. Also, the matching may be Pareto inefficient. Next, we will analyze these issues. SUDO roommate rule for b^2 type beds (hereafter, SUDO-2RR) is as follows:

3.1.1 The SUDO Roommate Rule for b^2 Type Beds (SUDO-2RR)

SUDO-2RR works in two stages. The first stage separate the set of students into disjoint subsets and the second stage assigns the students to their actual rooms.

Stage 1: First stage is the formation of pairs and singles based on priorities and declared roommate lists.

Step 1: Start with student i in M^2 with $\theta^i_{M^2} = 1$ and consider R_i^1.

Case 1 $[R_i^1 = \emptyset]$: Leave i as single.

Case 2 $[R_i^1 \neq \emptyset]$: Consider R_j^1 where $R_i^1 = \{j\}$.

Case 2.1 $[i \notin R_j^1]$: Leave i as single.

Case 2.1 [$i \in R^1_j$]: Make i and j a pair.

In general at

Step k: Consider student i with $\theta^i_{M^2} = k$ and consider R^1_i.

Case 1 [i is in a pair]: Leave i with his roommate.

Case 2 [$R^1_i = \emptyset$]: Leave i as single.

Case 3 [$R^1_i \neq \emptyset$]: Consider R^1_j where $R^1_i = \{j\}$.

Case 3.1 [$i \notin R^1_j$]: Leave i as single.

Case 3.1 [$i \in R^1_j$]: Make i and j a pair.

The algorithm terminates at Step $|M^2|$.

In fact, the outcome of this first stage is a matching of the students in M^2. In this matching γ, for any student $i \in M^2$, either $\gamma(i) = i$ which means student i is single or $\gamma(i) = j$ where $j \neq i$ and $\gamma(j) = i$ which means students i and j are roommates.

Stage 2: Second stage is the assignment of actual rooms based on room order information, priorities and the first stage's outcome.

Associate a counter to each room r^2_l in D^2_m as follows: $c_{o^2_m(r^2_l)}$ keeps track of how many beds are still available in room r^2_l. Initially each counter is equal to 2.

Step 1: Start with student i in M^2 with $\theta^i_{M^2} = 1$. Assign i to room $r^2_l \in D^2_m$ with $o^2_m(r^2_l) = 1$.

Case 1 [$\gamma(i) = i$]: Depending on the room i is assigned, the associated room counter is decreased by one.

Case 2 [$\gamma(i) = j$]: Assign j to the room i is assigned. The associated room counter is decreased by two.

Step 2: Consider student i in M^2 with $\theta^i_{M^2} = 2$.

Case 1 [i is assigned a room]: Leave i in his room with his roommate.

Case 2 [i is not assigned a room]: Assign i to room $r^2_l \in D^2_m$ with $o^2_m(r^2_l) = 2$.

Case 2.1 [$\gamma(i) = i$]: Depending on the room i is assigned, the associated room counter is decreased by one.

Case 2.2 [$\gamma(i) = j$]: Assign j to the room i is assigned. The associated room counter is decreased by two.

In general at

Step k: Consider student i in M^2 with $\theta^i_{M^2} = k$.

Case 1 [$\forall r^2_l \in D^2_m$, $c_{o^2_m(r^2_l)} = 0$]: Terminate the algorithm.

Case 2 [$\exists r^2_l \in D^2_m$ such that $c_{o^2_m(r^2_l)} \neq 0$]:

Case 2.1 [i is assigned a room]: Leave i in his room with his roommate.

Case 2.2 [i is not assigned a room and $\exists r^2_l \in D^2_m$ such that $c_{o^2_m(r^2_l)} = 2$]: Assign i to room $r^2_l \in D^2_m$ where $c_{o^2_m(r^2_l)} = 2$ and $\forall r^2_p \in D^2_m$ with $c_{o^2_m(r^2_p)} = 2$, $o^2_m(r^2_l) \leq o^2_m(r^2_p)$.

Case 2.2.1 [$\gamma(i) = i$]: Depending on the room i is assigned, the associated room counter is decreased by one.

Case 2.2.2 [$\gamma(i) = j$]: Assign j to the room i is assigned. The associated room counter is decreased by two.

Case 2.3 [i is not assigned a room and $\nexists r^2_l \in D^2_m$ such that $c_{o^2_m(r^2_l)} = 2$]: Assign i to room $r^2_l \in D^2_m$ where $c_{o^2_m(r^2_l)} = 1$ and $\forall r^2_p \in D^2_m$ with $c_{o^2_m(r^2_p)} = 1$, $o^2_m(r^2_l) \leq o^2_m(r^2_p)$. Depending on the room i is assigned, the associated room counter is decreased by one.

The algorithm terminates when there are no students left to consider or all the counters are equal to zero.

SUDO-2RR's main objective is to assign the students to the rooms, not to match the students to students. Because of this, the members of the pairs that may appear in the outcome of the first stage can be separated in the second stage. However, this causes the outcome of SUDO-2RR being unstable and Pareto inefficient. An example is as follows.

Example 17 *Consider a b^2-RP with $M^2 = \{i_1, i_2, ..., i_6\}$ where for any $i_k \in M^2$, $\theta^{i_k}_{M^2} = k$ and the following dichotomous preferences:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$
$\{i_2\}$	$\{i_3\}$	$\{i_6\}$	$\{i_5\}$	$\{i_4\}$	$\{i_1\}$
$M^2 \backslash \{i_2\}$	$M^2 \backslash \{i_3\}$	$M^2 \backslash \{i_6\}$	$M^2 \backslash \{i_5\}$	$M^2 \backslash \{i_4\}$	$M^2 \backslash \{i_1\}$

The algorithm used in the first stage makes the pair $\{i_4, i_5\}$ and leaves the other students single. However, the algorithm used in the second stage assigns students i_1, i_4 to the first room, students i_2, i_5 to the second room and students i_3, i_6 to the third room. Therefore, SUDO-2RR produces matching μ which matches i_1 and i_4, i_2 and i_5, i_3 and i_6. However, there is a blocking pair $\{i_4, i_5\}$ for this matching. Hence, μ is unstable.

Also, another matching μ', which matches i_1 and i_2, i_4 and i_5, i_3 and i_6, Pareto dominates μ. Therefore, SUDO-2RR is also Pareto inefficient.

However, these failures of SUDO-2RR disappear if we modify it as follows:

3.1.2 Stable and Pareto Efficient Roommate Rule for b^2-RP (b^2-RR)

We denote this modified version of SUDO-2RR by b^2-RR. In this rule, we use SUDO's first stage and second stage algorithms as they are. However, we introduce two new algorithms for the first stage. The rule b^2-RR works as follows:

Run SUDO-2RR's first stage algorithm. After this algorithm terminates, if all the students have a pair, then continue to the second stage. However, if there are some single students, then run the following algorithm before going to the second stage:

Second Algorithm:

Separate students (who are single in the outcome of the first stage) from M^2 and make the set S from these single students. Order the students in S by θ_S induced by ordering function θ as follows: For every $i \in S$, $\theta^i_S = \theta^i_{M^2} - |\{j \in (M^2 \backslash S) | \theta^i_{M^2} > \theta^j_{M^2}\}|$.

Step 1: Start with student i in S with $\theta^i_S = 1$ and consider R^1_i.

Case 1 [$R^1_i = \emptyset$]: Leave i as single.

Case 2 [$R^1_i = \{j\}$]: Make i and j a pair.

In general at

Step k: Consider student i in S with $\theta^i_S = k$.

Case 1 [i already has a pair]: Leave i with his mate.

Case 2 [i does not have a pair]:

Case 2.1 $[R_i^1 = \emptyset]$: Leave i as single.

Case 2.2 $[R_i^1 = \{j\}]$: Make i and j a pair.

The algorithm terminates after step $|S|$.

Now, if all the students have a pair in the outcome of this algorithm, then continue to the second stage. However, if there are still some single students, then run the following algorithm before going to the second stage:

Third Algorithm:

Separate students (who are single in the outcome of the second algorithm) from S and make the set SS from these single students. Order the students in SS by θ_{SS} induced by ordering function θ as follows: For every $i \in SS$, $\theta^i_{SS} = \theta^i_S - |\{j \in (S \backslash SS) | \theta^i_S > \theta^j_S\}|$.

Step 1: Make students i, j in SS a pair where $\theta^i_{SS} = 1$ and $\theta^j_{SS} = 2$.

In general at

Step k: Make students i, j in SS a pair where $\theta^i_{SS} = 2k - 1$ and $\theta^j_{SS} = 2k$.

The algorithm terminates after step $\frac{|SS|}{2}$.

After the first stage, all students must have a pair. Because of this, no pair can be splitted off in the second stage. Therefore, every member of a pair in the outcome of the first stage will be assigned the same room. Hence, the outcome of the second stage will be identical to the first stage's.

We had discussed that SUDO-2RR may produce unstable and Pareto inefficient matchings for instances of b^2-RPs. Next, we will show that b^2-RR eliminates these shortcomings.

Proposition 18 *The rule b^2-RR produces a stable matching for any problem b^2-RP with dichotomous preference profile.*

Proof. If b^2-RR's outcome μ is produced by the first algorithm, then it must be the case that each student i's roommate $\mu(i)$ is already in his roommate list. That is, $\mu(i) \in R_i^1$. Therefore, no student i can find another student j where $jP_i\mu(i)$. However, if it is produced by the second or third algorithm, then there must be a student i where his current roommate

$\mu(i)$ is not in his roommate list. That is, $\mu(i) \notin R_i^1$. However, i can not find another student j to form a blocking pair. Otherwise, they must already be paired in the first algorithm. ■

Proposition 19 *The rule b^2-RR produces a Pareto efficient matching μ for any problem b^2-RP with dichotomous preference profile where for any $i \in M^2$, $|R_i^1| \leq 1$.*

Proof. Since at least one of the students in the pairs produced by the first or second algorithm is matched to his top choice, any other matching which changes these pairs will hurt at least one of these students. Therefore, any Pareto dominating matching μ' must contain the pairs produced by the first or second algorithm. On the other hand, for any two students $i, j \in SS$, $R_i^1 \neq \{j\}$. Hence, any student $i \in SS$ can not be better-off unless he is matched to student $j \in M^2 \backslash SS$ where $R_i^1 = \{j\}$. ■

Note that if the top choice indifference classes can contain more than one student, then b^2-RR may produce Pareto inefficient matchings. An example is as follows.

Example 20 *Consider a b^2-RP with $M^2 = \{i_1, i_2, i_3, i_4\}$ where for any $i_k \in M^2$, $\theta_{M^2}^{i_k} = k$ and the following dichotomous preferences:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$
$\{i_2, i_4\}$	$\{i_1, i_3\}$	$\{i_2\}$	$\{i_1\}$
$\{i_3\}$	$\{i_4\}$	$\{i_1, i_4\}$	$\{i_2, i_3\}$

For this problem, b^2-RR produces stable matching μ which makes i_1, i_2 a pair and i_3, i_4 a pair. However, it is Pareto dominated by matching μ' which makes i_1, i_4 a pair and i_2, i_3 a pair.

Next, we analyze the roommate problem where the rooms have four beds. Therefore, for any student there will be more than one roommate in an outcome of a roommate formation. As far as we know, this will be the first analysis of such a problem.

3.2 Roommate Problem for b^4 Type Beds

In a roommate problem for b^4 type beds, there is a set of students denoted by M^4 which has a finite cardinality divisible by 4. Each student i in M^4 is assumed to have a **preference relation** R_i on M^4. We assume that these preference relations are complete, reflexive and transitive. For the associated **strict preference relation** and **indifference relation**, we will use P_i and I_i respectively.

As before, R_i^1 denotes student i's first choice, R_i^2 denotes his second choice, and so on. For any $i \in M^4$, we let R_i such that for any $j \in M^4$, jR_ii. The set of all preference relations on M^4 is $\mathcal{R}$. A vector consisting of every student's preference relations is called a **preference profile** and is denoted by $R = (R_1, ..., R_{|M^4|})$. The set of all preference profiles is denoted by $\mathcal{R}^{M^4}$.

For each student i, a binary relation $\mathcal{P}_i$ on $(M^4)^3$ induced by R_i is defined as follows: for any $\{a, b, c\}, \{j, k, l\} \in (M^4)^3$, $\{a, b, c\}\mathcal{P}_i\{j, k, l\}$ if there exist $p \in \{a, b, c\}$, $q \in \{j, k, l\}$ such that pR_iq and there exist $r \in \{a, b, c\}\backslash\{p\}$, $s \in \{j, k, l\}\backslash\{q\}$ such that rR_is and for $t \in \{a, b, c\}\backslash\{p, r\}$, $u \in \{j, k, l\}\backslash\{q, s\}$ it is the case that tR_iu and at least one of these relations is strict. Indeed, $\mathcal{P}_i$ is an asymmetric and negatively transitive binary relation. We call $\mathcal{P}_i$ as a **group preference relation**. For the negation of $\mathcal{P}_i$, we will use $\tilde{\mathcal{P}}_i$. The associated **group indifference relation** $\mathcal{I}_i$ is defined as follows: for any $\{a, b, c\}, \{j, k, l\} \in (M^4)^3$, $\{a, b, c\}\mathcal{I}_i\{j, k, l\}$ if and only if $\{a, b, c\}\tilde{\mathcal{P}}_i\{j, k, l\}$ and $\{j, k, l\}\tilde{\mathcal{P}}_i\{a, b, c\}$.

For student i, $\mathcal{P}_i^1$ denotes student i's first group choice, $\mathcal{P}_i^2$ denotes his second group choice, and so on. A vector consisting of every student's group preference relations is called a **group preference profile** and is denoted by $\mathcal{P} = (\mathcal{P}_1, ..., \mathcal{P}_{|M^4|})$.

There is an asymmetric, negatively transitive and weakly connected binary relation on M^4 denoted by θ_{M^4}. In fact, this relation is induced by the priority ordering θ defined on the set of all students N in the previous section. For any student $i \in M^4$, $\theta_{M^4}^i$ is the order of this student and it is defined as follows: $\theta_{M^4}^i = \theta^i - |\{j \in (N\backslash M^4)|\theta^i > \theta^j\}|$.

There is a set of type4 rooms D_m^4 where $|D_m^4| = \frac{|M^4|}{4}$. An associated function o_m^4 gives the order of each room in D^4_m. That is, $o^4_m : D^4_m \to \{1, 2, ..., \frac{|M^4|}{4}\}$. We call this function **room ordering**.

The **roommate problem for b^4 type beds** (hereafter, b^4-RP) is a vector consisting of the set of students M^4, the priority ordering θ_{M^4}, the set of type4 rooms D^4_m, the room ordering function o^4_m and a preference profile R: $(M^4, \theta_{M^4}, D^4_m, o^4_m, R)$. However, since in our model only R can be different between any two different b^4-RPs, with abuse of notation we will use R also for a b^4-RP. By the same reasoning, the set of all b^4-RPs is denoted by $\mathcal{R}^{M^4}$.

An outcome of a b^4-RP is a partition of M^4 into $|M^4|/4$ disjoint groups. We call this outcome a **matching** and denote it by μ. In fact, a matching μ is a one-to-one mapping from M^4 into $(M^4)^3$ such that for each $i \in M^4$, $|\mu(i)| = 3$ and for every $i, j, k \in M^4$, if $j \in \mu(i)$ and $k \in \mu(i)$ then $j \in \mu(k)$ and $k \in \mu(j)$. Each student in such a group is called the **roommate** of the other students in this group. The set of all matchings for a b^4-RP is denoted by $\mathcal{M}^4$.

Four students $\{i, j, k, l\}$ such that $i \neq j \neq k \neq l$ **block** a matching μ if $\{j, k, l\}\mathcal{P}_i\mu(i)$, $\{i, k, l\}\mathcal{P}_j\mu(j)$, $\{i, j, l\}\mathcal{P}_k\mu(k)$ and $\{i, j, k\}\mathcal{P}_l\mu(l)$. We call such a group as a **blocking group**. In a matching μ, if there are no blocking groups, we say that μ is *stable*. A matching $\mu \in \mathcal{M}^4$ is *Pareto efficient* if there does not exist any $\mu' \in \mathcal{M}^4$ such that for each $i \in M^4$, $\mu(i)\tilde{\mathcal{P}}_i\mu'(i)$ and there exits at least one $i \in M^2$ where $\mu'(i)\mathcal{P}_i\mu(i)$.

A *roommate rule* for b^4-RPs (hereafter, 4-RR) $\mathcal{F}$ is a systematic procedure that produces a matching for each b^4-RP. That is, $\mathcal{F} : \mathcal{R}^{M^4} \to \mathcal{M}^4$. A 4-RR $\mathcal{F}$ is Pareto efficient if for each $R \in \mathcal{R}^{M^4}$, $\mathcal{F}(R)$ is Pareto efficient.

Since the cardinality of the set of students is finite, for any R there must be a Pareto efficient matching μ. However, for some R, there may not exist a stable matching as the following example suggests:

Example 21 *Consider a b^4-RP with $M^4 = \{i_1, i_2, ..., i_8\}$ where for any $i_k \in M^4$, $\theta^{i_k}_{M^4} = k$ and the following strict preferences:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$	$\mathbf{i}_7$	$\mathbf{i}_8$
i_2	i_3	i_4	i_5	i_1	i_7	i_8	i_6
i_3	i_4	i_5	i_1	i_2	i_8	i_6	i_7
i_4	i_5	i_1	i_2	i_3	i_1	i_1	i_1
i_5	i_1	i_2	i_3	i_4	i_2	i_2	i_2
i_6	i_6	i_6	i_6	i_6	i_3	i_3	i_3
i_7	i_7	i_7	i_7	i_7	i_4	i_4	i_4
i_8	i_8	i_8	i_8	i_8	i_5	i_5	i_5
i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8

From these preferences, we can construct the first five students' preferences over 3-student groups among each other as follows:

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$
$\{i_2, i_3, i_4\}$	$\{i_3, i_4, i_5\}$	$\{i_4, i_5, i_1\}$	$\{i_5, i_1, i_2\}$	$\{i_1, i_2, i_3\}$
$\{i_2, i_3, i_5\}$	$\{i_3, i_4, i_1\}$	$\{i_4, i_5, i_2\}$	$\{i_5, i_1, i_3\}$	$\{i_1, i_2, i_4\}$
$\{i_2, i_4, i_5\}$	$\{i_3, i_5, i_1\}$	$\{i_4, i_1, i_2\}$	$\{i_5, i_2, i_3\}$	$\{i_1, i_3, i_4\}$
$\{i_3, i_4, i_5\}$	$\{i_4, i_5, i_1\}$	$\{i_5, i_1, i_2\}$	$\{i_1, i_2, i_3\}$	$\{i_2, i_3, i_4\}$

Five possible matchings where each makes four of these five students roommate to each other are as below:

$$\mu_1 = \{\{i_1, i_2, i_3, i_4\}, \{i_5, i_6, i_7, i_8\}\},\ \mu_2 = \{\{i_5, i_2, i_3, i_4\}, \{i_1, i_6, i_7, i_8\}\},\ \mu_3 = \{\{i_5, i_1, i_3, i_4\}, \{i_2, i_6, i_7, i_8\}\}$$

$$\mu_4 = \{\{i_5, i_1, i_2, i_4\}, \{i_3, i_6, i_7, i_8\}\},\ \mu_5 = \{\{i_5, i_1, i_2, i_3\}, \{i_4, i_6, i_7, i_8\}\}$$

However, $\{i_5, i_2, i_3, i_4\}$ block μ_1, $\{i_5, i_1, i_3, i_4\}$ block μ_2, $\{i_5, i_1, i_2, i_4\}$ block μ_3, $\{i_5, i_1, i_2, i_3\}$ block μ_4, and $\{i_1, i_2, i_3, i_4\}$ block μ_5. Therefore, these matchings turn to one another and the following cycle occurs:

$$\mu_1 \to \mu_2 \to \mu_3 \to \mu_4 \to \mu_5 \to \mu_1$$

On the other hand, the other possible matchings are also not stable. These matchings and corresponding blocking groups are as follows. Below $\{x, y, z\} = \{i_6, i_7, i_8\}$.

1. $\mu = \{\{i_1, i_2, x, y\}, \{i_3, i_4, i_5, z\}\}$ is blocked by $\{i_2, i_3, i_4, i_5\}$ and μ_2 occurs.
2. $\mu = \{\{i_1, i_3, x, y\}, \{i_2, i_4, i_5, z\}\}$ is blocked by $\{i_1, i_2, i_4, i_5\}$ or $\{i_2, i_3, i_4, i_5\}$ and μ_4 or μ_2 occurs respectively.
3. $\mu = \{\{i_1, i_4, x, y\}, \{i_2, i_3, i_5, z\}\}$ is blocked by $\{i_1, i_2, i_3, i_5\}$ or $\{i_2, i_3, i_4, i_5\}$ and μ_5 or μ_2 occurs respectively.
4. $\mu = \{\{i_1, i_5, x, y\}, \{i_2, i_3, i_4, z\}\}$ is blocked by $\{i_1, i_2, i_3, i_4\}$ and μ_1 occurs.
5. $\mu = \{\{i_2, i_3, x, y\}, \{i_1, i_4, i_5, z\}\}$ is blocked by $\{i_1, i_3, i_4, i_5\}$ and μ_3 occurs.
6. $\mu = \{\{i_2, i_4, x, y\}, \{i_1, i_3, i_5, z\}\}$ is blocked by $\{i_1, i_2, i_3, i_5\}$ or $\{i_1, i_3, i_4, i_5\}$ and μ_5 or μ_3 occurs respectively.
7. $\mu = \{\{i_2, i_5, x, y\}, \{i_1, i_3, i_4, z\}\}$ is blocked by $\{i_1, i_2, i_3, i_4\}$ or $\{i_1, i_3, i_4, i_5\}$ and μ_1 or μ_3 occurs respectively.
8. $\mu = \{\{i_3, i_4, x, y\}, \{i_1, i_2, i_5, z\}\}$ is blocked by $\{i_1, i_2, i_4, i_5\}$ and μ_4 occurs.
9. $\mu = \{\{i_3, i_5, x, y\}, \{i_1, i_2, i_4, z\}\}$ is blocked by $\{i_1, i_2, i_3, i_4\}$ or $\{i_1, i_2, i_4, i_5\}$ and μ_1 or μ_4 occurs respectively.
10. $\mu = \{\{i_4, i_5, x, y\}, \{i_1, i_2, i_3, z\}\}$ is blocked by $\{i_1, i_2, i_3, i_5\}$ and μ_5 occurs.

But we know that $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5$ are not stable matchings. Therefore, there is not any stable matching for this example.

In the previous subsection, we see that when the preference profile is dichotomous, then there exist a stable matching for a b^2-RP. In a b^2-RP, every student is to be matched to a student. However, in a b^4-RP, each student is to be matched to a 3-student group. Therefore, a natural question arises as follows. In a b^4-RP, if each student's group preference relation is dichotomous, can there exist a stable matching for this problem? The following proposition answers this question affirmatively.

Proposition 22 *For a b^4-RP, if the group preference profile $\mathcal{P}$ is dichotomous, then there exist a stable matching for this problem.*

The proof is similar to the proof of Chung's (2000) proposition since the structure of these two problems are the same.

Proof. Let μ_1 be an arbitrary matching. Suppose that μ_1 has a blocking group $\{a, b, c, d\}$. (If no blocking groups exist, then we are done.) That is, $\{b, c, d\}\mathcal{P}_a\mu_1(a)$, $\{a, c, d\}\mathcal{P}_b\mu_1(b)$, $\{a, b, d\}\mathcal{P}_c\mu_1(c)$ and $\{a, b, c\}\mathcal{P}_d\mu_1(d)$. Make a, b, c, d a group. Now, the set $[\mu_1(a) \cup \mu_1(b) \cup \mu_1(c) \cup \mu_1(d)]\backslash\{a, b, c, d\}$ has cardinality divisible by four. Make groups from the students in this set by using their priority ordering. Explicitly, make the first four students a group, the second four students another group and so on. Let other groups in μ_1 be the same. Now, we have another matching μ_2. Note that, from now on a, b, c, d can never be in another blocking group since the preferences are dichotomous and therefore $\nexists A \in (M^4)^3$ such that $A\mathcal{P}_a\{b, c, d\}$, $\nexists B \in (M^4)^3$ such that $B\mathcal{P}_b\{a, c, d\}$, $\nexists C \in (M^4)^3$ such that $C\mathcal{P}_c\{a, b, d\}$ and $\nexists D \in (M^4)^3$ such that $D\mathcal{P}_d\{a, b, c\}$. Proceed for i_2 and the other matchings that may appear in the same way. Since there is a finite number of students, after a finite step we will get the desired stable matching. ■

In fact, a dichotomous group preference profile can be constructed from the students' preferences when these preferences are dichotomous and each top choice indifference class can contain at most one student. It is because for each student i any 3-student group A will be in his first group choice $\mathcal{P}_i^1$ if A contains R_i^1 and any 3-student group B will be in his second group choice $\mathcal{P}_i^2$ if B does not contain R_i^1. Therefore, the following corollary applies.

Corollary 23 *For a b^4-RP, if the preference profile R is dichotomous and for each $i \in M^4$, $|R_i^1| \leq 1$, then there exist a stable matching for this problem.*

As it is done for the problem b^2-RP, to assign the students to the rooms with their friends, SUDO asks each student in M^4 to declare his roommate list. Therefore, each student classifies every other student into two groups. Hence, each student has dichotomous preferences over the set of students. Also, SUDO restricts each student, who will get a b^4 type bed, to declare at most three students in his roommate list. Therefore, the cardinality of each student's top choice indifference class is at most three.

For this problem b^4-RP, SUDO uses an algorithm to assign the students to the rooms. However, we do not know whether a stable matching always exists for this problem. But, on the other hand, SUDO's roommate algorithm may produce an unstable matching even when there exist a stable matching for it. Also, the outcome of the SUDO algorithm may be Pareto inefficient in some cases. Before analyzing SUDO's algorithm, we will state the following propositions which guarantee the existence of a stable matching for a b^4-RP under certain assumptions.

Proposition 24 *For a b^4-RP, if the preference profile $\mathcal{R}$ is dichotomous and for each $i \in M^4$, $|R_i^1| \leq 2$, then there exist a stable matching for this problem.*

For the proof, we will use a similar approach to the Roth-Vande Vate (1990) random paths to stability algorithm. Our aim is to show the existence of a 4-student group which can never be broken by any subgroup of its members. Then, deductively we will get the desired matching.

Proof. Start with a random matching. If there does not exist any blocking group, then we are done. If not, then there exists at least one blocking group. Let this group be $G = \{a, b, c, d\}$. Observe that none of the students by himself can break this group since at least one of his top choices is already a roommate of him and outside of this group, there can be at most one student in his top choice. For the blocking group G, there can be the following cases:

Case 1 $[|\{i \in G : R_i^1 \subset G\}| \geq 3]$: The group can not be broken since all the students, whose top choice class is a subset of G, will never break this group. On the other hand, the student, whose top choice class is not a subset of G, can not break tis group by himself by the above reasoning.

Case 2 $[b \in R_a^1 \wedge c \in R_b^1 \wedge d \in R_c^1 \wedge a \in R_d^1]$: The group can not be broken since for any member of a subgroup which breaks the group, there can be at most one student in his top choice outside of the group.

Case 3 $[(c \in R_a^1 \cap R_b^1) \wedge \{a, b\} = R_c^1]$: The group can not be broken since no two or three of the students in the group can break it. It is because c already has his both top choices in the group and therefore he will not break it. But then, none of the other students will break the group since outside of this group, there can be at most one student in their top choice classes.

Case 4 $[b \in R_a^1 \wedge c \in R_b^1 \wedge a \in R_c^1 \wedge d \notin R_a^1 \cup R_b^1 \cup R_c^1]$: The group may be broken by a, b, c and e where $e \in R_a^1 \cap R_b^1 \cap R_c^1$. However, now for group $\{a, b, c, e\}$ it is Case 1 and therefore it can not be broken.

Case 5 $[b \in R_a^1 \wedge a \in R_b^1 \wedge (c, d \notin R_a^1 \cup R_b^1)]$: The group may be broken by a, b, e and f where $R_a^1 \subset \{b, e, f\}$ and $R_b^1 \subset \{a, e, f\}$. Now for this new blocking group $\{a, b, e, f\}$ if it is one of the above first three cases, then this group can not be broken. However, if it is Case 4, then we know that this group may be broken, but then there must be another group which can not be broken. But, if it is again Case 5, then $\{a, b, e, f\}$ may be broken by e, f, g and h where $R_e^1 \subset \{f, g, h\}$ and $R_f^1 \subset \{e, g, h\}$. For the worst-case scenario, there may occur a sequence of these blocking groups where each group possesses Case 5. However, since there is a finite number of students, this sequence should terminate after a finite number of steps. So after the final step, we would have a group which can not be broken.

Since presence of an unbreakable group in the problem would not affect the decision of the other students, we can separate this group from the rest of the problem. But then, we will have a smaller problem which owns the same structure as the original one. Since there is a finite number of students, by deductive reasoning after a finite number of steps, we will get the desired partition of the set of students. ■

When we observe the roommate lists of students in a real-life roommate problem, it is very likely to see that when student j is in student i's roommate list, then i is also in j's roommate list. It is because when i decides to add j to his list, i should already know j's decision about adding i to his list. Then, if i is not in j's list, he will probably think that he has no right to add j to his list. If we apply this assumption to b^4-RPs where roommate lists can contain at most three students (as SUDO requires), then by the following proposition we see that there always exist stable matchings for them.

Proposition 25 *For a b^4-RP, if the preference profile R is dichotomous, for each $i \in M^4$, $|R_i^1| \leq 3$ and for each $i, j \in M^4$, $j \in R_i^1$ if and only if $i \in R_j^1$, then there exist a stable matching for this problem.*

The proof is given in the Appendix.

SUDO may want to adopt the procedure used for the proof of *Proposition 25* to form the roommate groups. However, the actual dichotomous preference profile of the students may not own the assumptions in the statement of this proposition. Indeed, this type of roommate list declaration can be caused by coordination problems. But, SUDO can solve this problem as follows:

Allow every student to declare any subset of the students except himself as his roommate list. Then, for each student i, remove every student j, who does not declare i in his roommate list, from i's roommate list. Then, for the first student i, remove every student except the first three ones from i's roommate list. Except from these first three students' lists, remove i from every other students' list. Then, for the second student, proceed as it is done for the first student and so on. At the end, the kind of preference profile in *Proposition 25* will be constructed.

Next, we will introduce SUDO's roommate rule for b^4 type beds (hereafter, SUDO-4RR). Then, we will investigate its shortcomings.

3.2.1 The SUDO Roommate Rule for b^4 Type Beds (SUDO-4RR)

Similar to the rule SUDO-2RR, SUDO-4RR works in two stages. The first stage splits the set of students into disjoint subsets and the second stage assigns the students to their actual rooms.

Stage 1: First stage is the formation of singles, pairs, 3-student and 4-student groups based on priorities and declared roommate lists.

Step 1: Start with student i in M^4 with $\theta^i_{M^4} = 1$ and consider R^1_i.

Case 1 [$|R^1_i| = 0$]: Leave i as single. Remove him from all the other students' top choice classes.

Case 2 [$|R^1_i| = 1$]: Consider R^1_j where $R^1_i = \{j\}$.

Case 2.1 [$i \notin R^1_j$]: Leave i as single. Remove him from all the other students' top choice classes.

Case 2.1 [$i \in R^1_j$]: Make i and j a pair. Remove them from all the other students' top choice classes.

Case 3 [$|R^1_i| = 2$]: Consider R^1_j where $R^1_i = \{j, k\}$ and $\theta^j < \theta^k$.

Case 3.1 [$i \notin R^1_j$]: Consider R^1_k.

Case 3.1.1 [$i \notin R^1_k$]: Leave i as single. Remove him from all the other students' top choice classes.

Case 3.1.2 [$i \in R^1_k$]: Make i and k a pair. Remove them from all the other students' top choice classes.

Case 3.2 [$i \in R^1_j$]: Consider R^1_k.

Case 3.2.1 [$i \notin R^1_k$]: Make i and j a pair. Remove them from all the other students' top choice classes.

Case 3.2.2 [$i \in R^1_k \wedge j \notin R^1_k$]: Make i and j a pair. Remove them from all the other students' top choice classes.

Case 3.2.3 [$i \in R_k^1 \wedge j \in R_k^1 \wedge k \notin R_j^1$]: Make i and k a pair. Leave j as single. Remove i and k from all the other students' top choice classes.

Case 3.2.4 [$i \in R_k^1 \wedge j \in R_k^1 \wedge k \in R_j^1$]: Make i, j and k a 3-student group. Remove them from all the other students' top choice classes.

Case 4 [$|R_i^1| = 3$]: Consider R_j^1 where $R_i^1 = \{j, k, l\}$ and $\theta^j < \theta^k < \theta^l$.

Case 4.1 [$i \notin R_j^1$]: Proceed as Case 3.

Case 4.2 [$i \in R_j^1$]: Consider R_k^1.

Case 4.2.1 [$i \notin R_k^1$]: Consider R_l^1. Proceed as Case 3.2 for i, j, l.

Case 4.2.2 [$i \in R_k^1 \wedge j \notin R_k^1$]: Consider R_l^1. Proceed as Case 3.2 for i, j, l.

Case 4.2.3 [$i \in R_k^1 \wedge j \in R_k^1$]: Consider R_l^1.

Case 4.2.3.1 [$i \notin R_l^1$]: Make i, j and k a 3-student group. Remove them from all the other students' top choice classes.

Case 4.2.3.2 [$i \in R_l^1 \wedge j \notin R_l^1$]: Make i, j and k a 3-student group. Remove them from all the other students' top choice classes.

Case 4.2.3.3 [$i \in R_l^1 \wedge j \in R_l^1 \wedge k \notin R_l^1$]: Make i, j and k a 3-student group. Remove them from all the other students' top choice classes.

Case 4.2.3.4 [$i \in R_l^1 \wedge j \in R_l^1 \wedge k \in R_l^1$]: Make i, j, k and l a 4-student group. Remove them from all the other students' top choice classes.

In general at

Step k: Consider student i with $\theta^i_{M^4} = k$ and consider R_i^1.

Case 1 [i is not single]: Leave i in his group.

Case 2 [i is single]: Proceed as Step 1.

The algorithm terminates at Step $|M^4|$. Now, the set M^4 is separated into disjoint subsets where some of these subsets are singleton, some consist of two students, some consist of three students and some consist of four students.

Stage 2: Second stage is the assignment of actual rooms based on room order information, priorities and the first stage's outcome.

Associate a counter to each room r_l^4 in D_m^4 as follows: $c_{o_m^4(r_l^4)}$ keeps track of how many beds are still available in room r_l^4. Initially each counter is equal to 4.

Step 1: Start with student i in M^4 with $\theta^i_{M^4} = 1$. Assign i to room $r_l^4 \in D_m^4$ with $o_m^4(r_l^4) = 1$.

Case 1 [i is single]: The associated room counter is decreased by one.

Case 2 [i is not single]: Assign other members of the group to the room i is assigned. The associated room counter is decreased by the size of the group.

Step 2: Consider student i in M^4 with $\theta^i_{M^4} = 2$.

Case 1 [i is assigned a room]: Leave i in his room with his roommate(s).

Case 2 [i is not assigned a room]: Assign i to room $r_l^4 \in D_m^4$ with $o_m^4(r_l^4) = 2$.

Case 2.1 [i is single]: The associated room counter is decreased by one.

Case 2.2 [i is not single]: Assign other members of the group to the room i is assigned. The associated room counter is decreased by the size of the group.

In general at

Step k: Consider student i in M^4 with $\theta^i_{M^4} = k$.

Case 1 [$\forall r_l^4 \in D_m^4$, $c_{o_m^4(r_l^4)} = 0$]: Terminate the algorithm.

Case 2 [$\exists r_l^4 \in D_m^4$ such that $c_{o_m^4(r_l^4)} \neq 0$]:

Case 2.1 [i is assigned a room]: Leave i in his room with his roommate(s).

Case 2.2 [i is not assigned a room]: Assign i to room $r_l^4 \in D_m^4$ where $c_{o_m^4(r_l^4)} \neq 0$ and $\forall r_p^4 \in D_m^4$ with $c_{o_m^4(r_p^4)} \neq 0$, $o_m^4(r_l^4) \leq o_m^4(r_p^4)$.

Case 2.2.1 [i is single]: The associated room counter is decreased by one.

Case 2.2.2 [i is not single]: Assign other than i the first $(c_{o_m^4(r_l^4)} - 1)$ members of the group to the room i is assigned. The associated room counter is decreased by the number of students assigned in this step.

The algorithm terminates when there are no students left to consider or all the counters are equal to zero.

Like SUDO-2RR, SUDO-4RR's main objective is to assign the students to the rooms, not to match the students to students. Because of this, the members of the groups that may appear in the outcome of the first stage can be separated in the second stage. However, even the outcome of the first stage is stable and/or Pareto efficient, this separation may cause the outcome of SUDO-4RR being unstable and/or Pareto inefficient.

The following example is an instance of a b^4-RP described in *Corollary 23* (Preference profile is dichotomous and each student's top choice class can contain at most one student). Therefore, there exists at least one stable matching for it.

Example 26 *Consider a b^4-RP with $M^4 = \{i_1, i_2, ..., i_{12}\}$ where for any $i_k \in M^4$, $\theta^{i_k}_{M^4} = k$ and the following dichotomous preferences:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$	$\mathbf{i}_6$
i_2	i_3	i_1	i_5	i_6	i_4
$M^4\backslash\{i_2\}$	$M^4\backslash\{i_3\}$	$M^4\backslash\{i_1\}$	$M^4\backslash\{i_5\}$	$M^4\backslash\{i_6\}$	$M^4\backslash\{i_4\}$

$\mathbf{i}_7$	$\mathbf{i}_8$	$\mathbf{i}_9$	$\mathbf{i}_{10}$	$\mathbf{i}_{11}$	$\mathbf{i}_{12}$
i_8	i_9	i_7	i_2	i_6	i_7
$M^4\backslash\{i_8\}$	$M^4\backslash\{i_9\}$	$M^4\backslash\{i_7\}$	$M^4\backslash\{i_2\}$	$M^4\backslash\{i_6\}$	$M^4\backslash\{i_7\}$

For these preferences and priorities, SUDO-4RR's first stage algorithm leaves every student as single. The second stage algorithm assigns students i_1, i_4, i_7 and i_{10} to the first room, students i_2, i_5, i_8 and i_{11} to the second room and students i_3, i_6, i_9 and i_{12} to the third room. Therefore, SUDO-4RR produces matching $\mu = \{\{i_1, i_4, i_7, i_{10}\}, \{i_2, i_5, i_8, i_{11}\}, \{i_3, i_6, i_9, i_{12}\}\}$. However, there are three blocking groups $\{i_1, i_2, i_3, i_{10}\}, \{i_4, i_5, i_6, i_{11}\}$ and $\{i_7, i_8, i_9, i_{12}\}$ for this matching. Hence, μ is unstable.

On the other hand, a stable matching $\mu' = \{\{i_1, i_2, i_3, i_{10}\}, \{i_4, i_5, i_6, i_{11}\}, \{i_7, i_8, i_9, i_{12}\}\}$ Pareto dominates μ. Therefore, SUDO-4RR is Pareto inefficient.

The following example is an instance of a b^4-RP described in *Proposition 24* or *25*. Therefore, there exists at least one stable matching for it.

Example 27 *Consider a b^4-RP with $M^4 = \{i_1, i_2, ..., i_8\}$ where for any $i_k \in M^4$, $\theta_{M^4}^{i_k} = k$ and the following dichotomous preferences:*

$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$
$\{i_2, i_3\}$	$\{i_1, i_4\}$	$\{i_1, i_4\}$	$\{i_2, i_3\}$
$M^4 \backslash \{i_2, i_3\}$	$M^4 \backslash \{i_1, i_4\}$	$M^4 \backslash \{i_1, i_4\}$	$M^4 \backslash \{i_2, i_3\}$

$\mathbf{i}_5$	$\mathbf{i}_6$	$\mathbf{i}_7$	$\mathbf{i}_8$
$\{i_6, i_7\}$	$\{i_5, i_8\}$	$\{i_5, i_8\}$	$\{i_6, i_7\}$
$M^4 \backslash \{i_6, i_7\}$	$M^4 \backslash \{i_5, i_8\}$	$M^4 \backslash \{i_5, i_8\}$	$M^4 \backslash \{i_6, i_7\}$

For these preferences and priorities, SUDO-4RR's first stage algorithm makes pairs $\{i_1, i_2\}, \{i_3, i_4\}, \{i_5, i_6\}$ and $\{i_7, i_8\}$. The second stage algorithm assigns students i_1, i_2, i_5, i_6 to the first room and i_3, i_4, i_7, i_8 to the second room. Therefore, SUDO-4RR produces matching $\mu = \{\{i_1, i_2, i_5, i_6\}, \{i_3, i_4, i_7, i_8\}\}$. However, there are two blocking groups $\{i_1, i_2, i_3, i_4\}$ and $\{i_5, i_6, i_7, i_8\}$ for this matching. Hence, μ is unstable.

On the other hand, a stable matching $\mu' = \{\{i_1, i_2, i_3, i_4\}, \{i_5, i_6, i_7, i_8\}\}$ Pareto dominates μ. Therefore, SUDO-4RR produces Pareto inefficient matching for this example.

4 Conclusion

In this section, we list the open questions that one might pursue in a follow-up study.

One of the concerns of the SUDO roommate rules is the order of rooms while assigning students to their actual rooms. However, this room ordering is not based on any criteria. On the other hand, the groups may have actual preferences on the set of rooms. A group's preference order represents the group members' mutual interests on the rooms. Therefore, instead of assigning students to the rooms by using the randomly determined room orders, we can use these preferences and we can Pareto improve the solution.

However, there appears two major difficulties if we apply this approach to the problem. First, how can we construct a preference relation for each group? Second, how can we set a unique priority ordering for the set of groups?

Nevertheless, if we have a preference relation and a priority order for each group, then the problem just becomes the marriage problem. Here, men are the groups and women are the rooms or vice versa. As Abdulkadiroğlu and Sönmez (2003) notes, since there is a unique priority ordering for the groups, we can apply the serial dictatorship rule for this problem to create a stable, Pareto efficient and strategy-proof solution.

Also, the institution's main concern can be to increase the total welfare of the students. That is, the institution may want to have a partition of the set of students μ to maximize a social welfare function. For the particular social welfare function, $f(\mu) = \sum_{g \in \mu} \sum_{i \in g} |R_i^1 \cap g|$ the problem of finding such a partition (matching) is deeply investigated in graph theory and there are many algorithms which are used to maximize this function. Note that, the maximizer μ must be Pareto efficient. Otherwise, it can not be the maximum.

We show that for a b^4-RP, when the preferences are dichotomous and the top choice classes can contain at most 2 students, then there exists a stable matching. However, we do not know that a stable matching always exists when the top choice classes can contain at most 3 students or more. It will be interesting to find an upperbound for the cardinality of the top choice classes to guarantee the existence of a stable matching. On the other hand, we know that for a b^2-RP, there is no need to restrict the size of the top choice classes to have a stable matching. Therefore, a characterization of these upperbounds for the classes of roommate problems will also be interesting.

Finally, we did not investigate the implications of strategy-proofness for the b^4-RP problem. This also remains an open question for the future.

References

[1] Abdulkadiroğlu, A., Sönmez, T., (1998) "Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems", *Econometrica*, 66, 689-701.

[2] Abdulkadiroğlu, A., Sönmez, T., (1999) "House Allocation with Existing Tenants", *Journal of Economic Theory*, 88:2, 233-260.

[3] Abdulkadiroğlu, A., Sönmez, T., (2003) "School Choice: A Mechanism Design Approach", *American Economic Review*, 93:3, 729-747.

[4] Alkan, A., (1986) "Nonexistence of Stable Threesome Matchings", *Mathematical Social Sciences*, 16, 207-209.

[5] Chung, K., (2000) "On the Existence of Stable Roommate Matchings", *Games and Economic Behavior*, 33, 206-230.

[6] Gale, D., Shapley, S. L., (1962) "College Admissions and the Stability of Marriage", *American Mathematical Monthly*, 69:1, 9-15.

[7] Granot, D., (1984) "A Note on the Room-mates Problem and a Related Revenue Allocation Problem", *Management Science*, 30, 633-643.

[8] Gusfield, D., (1988) "The Structure of the Stable Roommate Problem: Efficient Representation and Enumeration of All Stable Assignments" *SIAM Journal on Computing*, 17, 742-769.

[9] Hylland, A., Zeckhauser, R., (1979) "The Efficient Allocations of Individuals to Positions", *Journal of Political Economy*, 87:2, 293-314.

[10] Irving, R. W., (1985) "An Efficient Algorithm for the Stable Room-mates Problem", *Journal of Algorithms*, 6, 577-595.

[11] Roth, A. E., Sotomayor, M., (1990) "Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis", *Econometric Society Monographs*, Cambridge: Cambridge Univ. Press.

[12] Roth, A. E., and Vande Vate, J. H. (1990). "Random Paths to Stability in Two-Sided Matching", *Econometrica*, 58, 1475–1480.

[13] Tan, J. J. M., (1991) "A Necessary and Sufficient Condition for the Existence of a Complete Stable Matching", *Journal of Algorithms*, 12, 154–178.

5 Appendix

Table 1

Year	Number of Applicants	Number of Beds
2005	2186	2072
2006	2424	2136
2007	2632	2414

Proof of Proposition 8:

Suppose that for a student selection problem P, GS-SDR produces a Pareto inefficient student selection σ. Then there must be at least one other student selection σ' which Pareto dominates σ. For such a Pareto dominating σ', there can not be any student i in N where $\sigma_i P_i \sigma'_i$ and there must be at least one student i in N where $\sigma'_i P_i \sigma_i$. Since for any i in N, $\sigma'_i = P_i^3 \Rightarrow P_i^3 = \varnothing$ and since for any i in N^s, $\sigma_i \neq P_i^3$, it must be the case that $\tilde{N}^s = N^s$ where $\tilde{N}^s$ is the set of selected students under σ'. Otherwise, there must be at least one student i in $N^s \backslash \tilde{N}^s$ and for that i, $\sigma_i P_i \sigma'_i$. But this will contradict with the supposition. Also, since the preferences are strict, then for every student i in N where $\sigma'_i \tilde{P}_i \sigma_i$, $\sigma'_i = \sigma_i$ and for every student i in N where $\sigma'_i P_i \sigma_i$, $\sigma'_i \neq \sigma_i$. $\tilde{M}^2, \tilde{M}^4, \tilde{F}^2, \tilde{F}^4$ refers to the sets of selected students according to their gender and bed types as in the definition of matching. Therefore, $\tilde{M}^2 \cup \tilde{M}^4 \cup \tilde{F}^2 \cup \tilde{F}^4 = \tilde{N}^s$.

Consider a strictly better off student i in this Pareto dominating selection σ'. Without loss of generality, assume that $g(i) = m$ and $\sigma_i = b^2$. By the above reasoning, it must be the case that $\sigma'_i = b^4$. But then, since all the rooms are reserved only for the selected students, there must be another student $j \in \tilde{N}^s$ such that $\sigma_j = b^4$ and $\sigma'_j = b^2$. Since the preferences are strict, then $\sigma'_j P_j \sigma_j$. Otherwise, since $b^2 \neq b^4$, $\sigma_j P_j \sigma'_j$ and this will contradict with the supposition that in this new selection σ', there is no student i in N where $\sigma_i P_i \sigma'_i$. Therefore, depending on $g(j)$, there is either sg-justified envy or og-justified envy in the outcome of GS-

SDR for that problem P since $\sigma_j P_i \sigma_i$ and $\sigma_i P_j \sigma_j$ and it is the fact that either $\theta^i < \theta^j$ or $\theta^j < \theta^i$ holds. By the Proposition 3, however, σ can not contain sg-justified envy. Therefore, it must be the case that $g(j) = f$.

However, to be a student selection, the Pareto dominating selection must satisfy the following conditions: $|\tilde{M}^2|$ and $|\tilde{F}^2|$ must be divisible by 2 and $|\tilde{M}^4|$ and $|\tilde{F}^4|$ must be divisible by 4. Therefore, other than student j there must be at least three female students k, l, h where $\sigma'_k = \sigma'_l = \sigma'_h = b^2$ and $\sigma_k = \sigma_l = \sigma_h = b^4$. Since these female students will be assigned to rooms in D^2, there must be at least three available beds for them. By this reason, other than student i, there must be at least three male students p, q, r where $\sigma'_p = \sigma'_q = \sigma'_r = b^4$ and $\sigma_p = \sigma_q = \sigma_r = b^2$. Now there can be two cases according to θ^i and θ^j.

Case 1: $\theta^i < \theta^j$. In σ, even i has higher priority than j, and $b^4 P_i b^2$, he can not be assigned to b^4 but j is assigned to b^4. Therefore, student t, who is considered at Step $\phi(t)$ when $c_4 = 4$ must be such that $\theta^t < \theta^i$ and $g(t) = g(j)$. Otherwise, i can be assigned to b^4. At this step, t is assigned to b^4. Meanwhile, male students p, q, r are also assigned to b^2 and they all prefer b^4 to b^2. Hence, it must be the case that $\theta^t < \theta^p$, $\theta^t < \theta^q$, and $\theta^t < \theta^r$. Otherwise at least one of students p, q, r can be assigned to b^4. Since female students k, l, h are assigned to b^4 and they all prefer b^2 to b^4, then it must be the case that at least three of students i, p, q, r have higher ranking than all students j, k, l, h. Otherwise, at least one of students j, k, l, h can be assigned to b^2. Therefore, student t has higher ranking than all these female students. That is $\theta^t < \theta^j$, $\theta^t < \theta^k$, $\theta^t < \theta^l$, and $\theta^t < \theta^h$. But then, this contradicts with the fact that after student t is assigned to b^4, there can be at most three other students who can be assigned to b^4. This is because after Step θ^t, $c_4 \leq 3$. Therefore this can not be the case.

Case 2: $\theta^i > \theta^j$. In σ, even j has higher priority than i, and $b^2 P_j b^4$, she can not be assigned to b^2 but i is assigned to b^2. Therefore, the student n, who is considered at Step θ^n when $c_2 = 2$, must be such that $\theta^n < \theta^j$ and $g(n) = g(i)$. Otherwise, j can be assigned to b^2. At this step, n is assigned to b^2. Meanwhile, since male students p, q, r are assigned to b^2, either they all must have higher ranking than n or two of them must have higher ranking than n and one of them must be n. Otherwise, j can be assigned to b^2. However, students

p, q, r all prefer b^4 to b^2 and they are assigned to b^2 while j is assigned to b^4. Hence, student t, who is considered at Step θ^t when $c_4 = 4$, must be such that $\theta^t < \theta^p, \theta^t < \theta^q, \theta^t < \theta^r$ and $g(t) = g(j)$. Otherwise, one of these male students p, q, r can be assigned to b^4. At this step, t is assigned to b^4. Meanwhile, female students k, l, h are also assigned to b^4 and they all prefer b^2 to b^4. Hence, it must be the case that at least three of students i, p, q, r have higher ranking than all the students j, k, l, h. Otherwise, at least one of j, k, l, h can be assigned to b^2. Therefore, student t has higher ranking than all these female students. That is $\theta^t < \theta^j$,$\theta^t < \theta^k$,$\theta^t < \theta^l$, and $\theta^t < \theta^h$. But then, this contradicts with the fact that after student t is assigned to b^4, then there can be at most three other students who can be assigned to b^4. This is because after Step θ^t, $c_4 \leq 3$. Therefore this can not also be the case.

Since there is no case left, there can not be any student selection in which at least one student is strictly better off without hurting someone in this selection. ■

Proof of Proposition 25: We will prove the proposition by construction in stages.

Stage 1: In this stage, we form 4-student groups which can not be broken. A 4-student group g_4 formed in this stage has the following property. For any student $i \in g_4$, $|R_i^1 \cap g_4| \geq 2$. We denote the set of such groups by $G_1^{g_4}$. We denote the set of students who are in a group in $G_1^{g_4}$ by A_1. The set of the remaining students is denoted by $A_1^n = M^4 \backslash A_1$. Any group g_4 in $G_1^{g_4}$ can not be broken. It is because any student has at least two of his top choices in the group and outside of this group there can be at most one student from his top choice class. Now, depending on $|A_1^n|$ there can be two cases:

Case 1 [$|A_1^n| = 0$]: We reach the desired matching $\mu = G_1^{g_4}$.

Case 2 [$|A_1^n| \neq 0$]: We should consider the following stage.

Stage 2: In this stage, we form 3-student groups from the students in A_1^n. A 3-student group g_3 formed in this stage has the following property. For any $i \in g_3$, $|R_i^1 \cap g_3| \geq 2$. We denote the set of such groups by $G_1^{g_3}$. We denote the set of students who are in a group in $G_1^{g_3}$ by B_1. The set of the remaining students is denoted by $B_1^n = M^4 \backslash B_1$. The groups formed in this stage are not complete yet. Now, depending on $|B_1^n|$ there can be two cases:

Case 1 [$|B_1^n| = 0$]: We will proceed to the following stage.

Stage 2.1: In this, stage we consider pair of 3-student groups $\{g_3, g_3'\}$ where there exists $i \in g_3$ and $j \in g_3'$ such that $j \in R_i^1$ and $i \in R_j^1$. Note that, if such a pair of groups exists, only a student in each group can be in the other student's top choice class. Otherwise, some of the students in these groups must be in the set A_1. Then we form three pair of students from these pair of 3-student groups in such a way. For the pair $\{g_3, g_3'\}$, we form the first pair $\{i, j\}$ from the students $i \in g_3$ and $j \in g_3'$ such that $j \in R_i^1$ and $i \in R_j^1$. We form the other pairs as $g_3 \backslash \{i\}$ and $g_3' \backslash \{j\}$. We denote the set of such pair of students by $G_1^{g_2}$. We denote the set of students who are in a pair in $G_1^{g_2}$ by C_1. We denote the set of 3-student groups who are not used to form pair of students by $G_{1'}^{g_3}$. The set of students in a group in $G_{1'}^{g_3}$ is denoted by B_2. Now, depending on $|B_2|$ there can be two cases:

Case 1.1 [$|B_2| = 0$]: We will proceed to the following stage.

Stage 2.1.1: In this stage, we form 4-student groups from the pairs in $G_1^{g_2}$. We denote the set of such groups by $G_2^{g_4}$. We do not impose any restriction on the formation of 4-student groups. A 4-student group g_4 can be randomly formed. Nevertheless, these groups can not be broken. It is because of the fact that any student has at least one of his top choices in the group and there can not be made any 4-student group from these students where each student has at least two of his top choices in the group. Otherwise, this 4-student group must be already in $G_1^{g_4}$. Therefore, we reach the desired matching $\mu = G_1^{g_4} \cup G_2^{g_4}$.

Case 1.2 [$|B_2| \neq 0$]: We will proceed to the following stage.

Stage 2.1.2: In this stage, like Stage 2.1, we form three pair of students from the pair of 3-student groups in $G_{1'}^{g_3}$. However, now we do not impose any restriction on the pair of 3-student groups. We denote the set of pair of students formed in this stage by $G_2^{g_2}$. Then we consider the following stage.

Stage 2.1.2': In this stage, like Stage 2.1.1, we form 4-student groups from the pairs in $G_1^{g_2} \cup G_2^{g_2}$. Again, we do not impose any restriction on the formation of groups. We denote the set of such groups by $G_2^{g_4}$. Again, by the same reasonings as in Stage 2.1.1, the 4-student groups can not be broken. Therefore, we reach the desired matching $\mu = G_1^{g_4} \cup G_2^{g_4}$.

Case 2 [$|B_1^n| \neq 0$]: We should consider the following stage.

Stage 3: In this stage, we form pair of students from the students in B_1^n. A pair g_2 formed in this stage has the following property. For any $g_2 = \{i, j\}$, $j \in R_i^1$ and $i \in R_j^1$. We denote the set of such pair of students by $G_1^{g_2}$. We denote the set of students who are in a pair in $G_1^{g_2}$ by C_1. We denote the set of the remaining students by $D_1 = B_1^n \backslash C_1$. Now, depending on $|D_1|$ there can be two cases:

Case 2.1 [$|D_1| = 0$]: We should consider the following stage.

Stage 3.1: This stage is identical to the Stage 2.1. Therefore, at the end of this stage, we have the following sets: the set of pair of students, which is formed from the students in B_1, denoted by $G_2^{g_2}$, the set of 3-student groups who are not used to form pair of students denoted by $G_2^{g_3}$ and the set of students in a group in $G_2^{g_3}$ is denoted by B_2. Now, depending on $|B_2|$ there can be two cases:

Case 2.1.1 [$|B_2| = 0$]: We will proceed to the following stage.

Stage 3.1.1: This stage is similar to the Stage 2.1.1. Now, however, we form 4-student groups from the pairs in $G_1^{g_2} \cup G_2^{g_2}$. We denote the set of such groups by $G_2^{g_4}$. Again, by the same reasonings as in Stage 2.1.1, the 4-student groups can not be broken. Therefore, we reach the desired matching $\mu = G_1^{g_4} \cup G_2^{g_4}$.

Case 2.1.2 [$|B_2| \neq 0$]: We will proceed to the following stage.

Stage 3.1.2: In this stage, we form singles and pair of students from the 3-student groups in $G_2^{g_3}$. For each group, we randomly select one student from the group and make him a single. We leave the other students is this group as a pair. At the end of the stage we have the following sets: the set of single students denoted by D_2 and the set of pair of students denoted by $G_3^{g_2}$. Then we consider the following stage.

Stage 3.1.2': In this stage, we form 4-student groups from the single students in D_2 and pair of students in $G_1^{g_2}$. A 4-student group g_4 formed in this stage has the following properties. First, each group is formed by two single students in D_2 and a pair of students in $G_1^{g_2}$. Second, for each group $\{i, j, k, l\}$ where $\{i, j\} \in G_1^{g_2}$ and $k, l \in D_2$, $k \in R_i^1 \backslash R_j^1$ if and only if $l \in R_j^1 \backslash R_i^1$ and $k \in (R_i^1 \cup R_j^1) \backslash (R_i^1 \cap R_j^1)$. We denote the set of such groups by $G_2^{g_4}$.

These groups can not be broken. It is because of the fact that students i and j have two of their top choices in the group and they can not increase these numbers being in another group since $|R_i^1 \cap R_j^1| \leq 2$. On the other hand, k or l has one of their top choices and they also can not increase these numbers being in another group. For instance, for k, the only way of increasing this number is being in a group consisting of his partners in the 3-student group in $G_1^{g_3}$. However, to form a 4-student group, these three students can not find a fourth student m who has at least one of his top choices in this group. It is because if $m \in B_1$, then these all students must be in $G_2^{g_2}$, but if $m \in G_2^{g_2}$ he has already one of his top choices in the pair in $G_2^{g_2}$. Therefore, he must have two of his top choices in the 4-student group. But then, these four students must already be in A_1. Now, the set of single students who are not in a 4-student group is D_3 and the set of pairs in $G_1^{g_2}$ but not used to form the 4-student groups is $G_{1'}^{g_2}$. Now, depending on $|D_3|$ and $|G_{1'}^{g_2}|$, there can be three cases:

Case 2.1.2.1 $[|D_3| = |G_{1'}^{g_2}| = 0]$: We will proceed to the following stage.

Stage 3.1.2'.1: This stage is similar to the Stage 2.1.1. Now, however, we form 4-student groups from the pairs in $G_2^{g_2} \cup G_3^{g_2}$. We denote the set of such groups by $G_3^{g_4}$. Again, by the same reasonings as in Stage 2.1.1, the 4-student groups can not be broken. Therefore, we reach the desired matching $\mu = G_1^{g_4} \cup G_2^{g_4} \cup G_3^{g_4}$.

Case 2.1.2.2 $[|G_{1'}^{g_2}| > |D_3| = 0]$: We will proceed to the following stage.

Stage 3.1.2'.2: This stage is similar to the Stage 2.1.1. Now, however, we form 4-student groups from the pairs in $G_{1'}^{g_2} \cup G_2^{g_2} \cup G_3^{g_2}$. We denote the set of such groups by $G_3^{g_4}$. Again, by the same reasonings as in Stage 2.1.1, the 4-student groups can not be broken. Therefore, we reach the desired matching $\mu = G_1^{g_4} \cup G_2^{g_4} \cup G_3^{g_4}$.

Case 2.1.2.3 $[|D_3| > |G_{1'}^{g_2}| = 0]$: We will proceed to the following stage.

Stage 3.1.2'.3: In this stage, we randomly form pair of students from the students in D_3. We denote the set of such pairs by $G_4^{g_2}$. Then we consider the following stage.

Stage 3.1.2'.3': This stage is similar to the Stage 2.1.1. Now, however, we form 4-student groups from the pairs in $G_2^{g_2} \cup G_3^{g_2} \cup G_4^{g_2}$. We denote the set of such groups by $G_3^{g_4}$. Again, by the same reasonings as in Stage 2.1.1, the 4-student groups can not be broken.

Therefore, we reach the desired matching $\mu = G_1^{g_4} \cup G_2^{g_4} \cup G_3^{g_4}$.

Case 2.2 [$|D_1| \neq 0$]: We should consider the following stage.

Stage 3.2: This stage is similar to the Stage 3.1.2'. Now, we form 4-student groups from the single students in D_1 and pair of students in $G_1^{g_2}$. We denote the set of such groups by $G_2^{g_4}$. By the same reasonings as in Stage 3.1.2', the 4-student groups can not be broken. The set of single students who are not in a 4-student group is now D_2 and the set of pairs in $G_1^{g_2}$ but not used to form the 4-student groups is now $G_2^{g_2}$. Depending on $|D_2|$ and $|G_2^{g_2}|$, there can be three cases:

Case 2.2.1 [$|D_2| = |G_2^{g_2}| = 0$]: This case is identical to the case where $|B_1^n| = 0$. Therefore, we have a stable matching for this case.

Case 2.2.2 [$|D_2| > |G_2^{g_2}| = 0$]: This case is identical to the case where $|D_3| > |G_{1'}^{g_2}| = 0$. Therefore, we have a stable matching for this case.

Case 2.2.3 [$|G_2^{g_2}| > |D_2| = 0$]: This case is identical to the case where $|D_1| = 0$. Therefore, we have a stable matching for this case.

The proof is done since for all the possible cases, we can find a stable matching. ■