Assignment and Pricing in Roommate Market

Pak Hay Chan
The Chinese University of Hong Kong, HK

Xin Huang
The Chinese University of Hong Kong, HK

Zhengyang Liu
Shanghai Jiao Tong University, China

Chihao Zhang
Shanghai Jiao Tong University, China

Shengyu Zhang
The Chinese University of Hong Kong, HK

Abstract

We introduce a *roommate market model*, in which $2n$ people need to be assigned to n rooms, with two people in each room. Each person has a valuation to each room, as well as a valuation to each of other people as a roommate. Each room has a rent shared by the two people living in the room, and we need to decide who live together in which room and how much each should pay. Various solution concepts on stability and envy-freeness are proposed, with their existence studied and the computational complexity of the corresponding search problems analyzed. In particular, we show that maximizing the social welfare is NP-hard, and we give a polynomial-time algorithm that achieves at least 2/3 of the maximum social welfare. Finally, we demonstrate a pricing scheme that can achieve envy-freeness for each room.

1 Introduction

Resource allocation is concerned with how to distribute available resources for different users. The practical applications of resource allocation include activities scheduling, project management, distribution of income in public finance, human resources management in strategic planning, etc..

One natural measurement for the quality of an outcome of a resource allocation mechanism is *social welfare*, which is the summation of utilities of all agents. However, the optimization of social welfare is not always the unique objective, and stability is usually another important concern. An allocation is stable if no coalition can devise new trades that made them better off. The stability is a central notion in cooperative game theory.

The stable matching problem has attracted a lots of researchers' attention since the seminal work (Gale and Shapley 1962). A line of work propose many variants and develop many efficient algorithms (Irving, Manlove, and Scott 2000; 2003; Baıou and Balinski 2000; Bansal, Agrawal, and Malhotra 2003; Malhotra 2004). Besides the theoretical interest, some algorithms have been successfully applied to matching systems such as National Resident Matching Program in the US and kidney exchange (Roth, Sönmez, and Ünver 2005).

One important extension of stable matching problem is stable roommate problem (Gale and Shapley 1962). In a given instance of the roommates problem, each of $2n$ participants ranks the others in strict order of preference. A matching is stable if there is no pair of participants such that both prefer the other to their current partners. Irving (Irving 1985) gave an efficient algorithm to decide whether there exists a stable solution and to output one if existing.

In a different vein, roommate market can be considered as a two sided market if we consider how to assign pairs of roommates into rooms based on their evaluations to rooms. A classic model for this is assignment game, in which each seller has a unique item to sell and each buyer has a unit demand (Shapley and Shubik 1971). The goal of assignment game is to compute a matching μ along with a vector of prices p. An desirable output is a price vector and a feasible assignment at which each seller maximizes revenues, each buyer maximizes net valuations, and markets clear.

In this paper, we propose a novel *roommate market model* that takes both room sharing and pricing into consideration. In this model, each participant $i \in I$ has a valuation v_{ir} of all the rooms r and, at the same time, has a happiness valuation h_{ij} of all the other participants j as the potential roommate. An *assignment* is set of triples $\mu = \{(i,j,r) : i,j \in I, r \in R\}$, with person i and j assigned to room r for each $(i,j,r) \in \mu$. A *rent vector* $p \in \mathbb{R}_+^{I \cup R}$ specifies the amount of rent p_i that each person $i \in I$ should pay and also the room rent p_r for each room r. If $(i,j,r) \in \mu$, then it holds that $p_i + p_j = p_r$. For each $(i,j,r) \in \mu$, the *utility* of person i is her happiness h_{ij} plus her evaluation v_{ir} of the room r, minus the price p_i she pays. The *social welfare* of assignment μ is defined as the summation of utilities of all participants plus the prices of all rooms, or equivalently the summation of the happiness of all participants and the evaluations of all rooms induced by the assignment μ. [1] For the problems studied in this paper v_{ir} and h_{ij} are given as input, and the task is to find a solution, including the assignment and rent vector, to satisfy certain desirable properties.

[1]Here we include the room prices in the social welfare because in canonical two-sided market (Roth and Sotomayor 1992), the social welfare includes the utility of agents from *both* sides. In our case, the utility of a room is its price, and the total price of all rooms can be viewed as the utility of the property owner (such as the university).

We also consider the scenario where the rent vector p is also part of input, corresponding to practical situations where the rent is set by university in line with the market value. Indeed, in most university housing, there is a fixed rent for all rooms (of the same type) and the two roommates in the same room pay the same rent. When the rent vector p is given, a solution to be found is simply an assignment.

Compared to the classic stable roommate model, ours quantifies the preference over potential roommates, allowing us to quantitatively define social welfare and to approximate its maximum value. In addition, we add the influence of the room valuation into consideration, making the model more applicable to certain scenarios.

Our Results

Our results focus on three major notions: social welfare, stability and envy-freeness.

We show that it is NP-hard to find a maximum social welfare assignment even if the valuations are all 0-1 values. On the positive side, we give an $O(n^3)$ time algorithm that can output an assignment with social welfare at least $2/3$ of the optimal.

For stability, we consider several solution concepts to cater different situations. A solution is *2-person stable* (2PS) if no pair of participants living in different rooms can both gain positive utility by swapping with each other. We show that it is NP-hard to find a solution satisfying this weaker property (2PS) even assuming the existence of such solutions. Actually, even deciding whether there exists a 2-person stable solution is also NP-hard.

One may go further and consider a practical situation that two participants want to switch rooms need to get consent of their roommates. This leads to the concept of 4-person stability: A solution is *4-person stable* (4PS) if there is no pair of rooms such that all 4 people in those rooms can increase their individual utility by swapping two residents originally living in different rooms. The complexity of finding a 4-person stable solution is quite different from that for 2-person stable solution. We show that there always exists a solution that is 4-person stable and demonstrate an $O(n^2)$ time algorithm that finds a solution that is 4-person stable. Furthermore, targeting on achieving both stability and good social welfare, we can find a 4PS solution with social welfare at least $2/3$ of the optimal, with the running time still a polynomial (albeit with an exponent larger than 2).

Finally, we consider envy-freeness, a property stronger than the stability mentioned above. A solution is *person envy-free* if each person cannot increase her utility by switching positions with any other person. It turns out that this is too strong a concept and such solution does not exist in general. A weaker concept is room envy-freeness where the two roommates living in a room are considered as a whole: A solution is *room envy-free* if any two people sharing a room cannot increase their total utility by switching to any other room. It turns out that a room envy-free solution always exists and can be found efficiently. One can also consider room stability, elaborated later.

The above results are summarized in Table 1.

Related Work

From the seminal paper (Gale and Shapley 1962), the stable matching has been widely studied. Many variants of stable matching has been proposed based on various practical applications (Irving, Manlove, and Scott 2000; 2003; Baıou and Balinski 2000; Bansal, Agrawal, and Malhotra 2003; Malhotra 2004).

Our roommate market model is closely related to two problems: 3-Dimensional Stable Matching (Knuth 1997) and Room Assignment-Rent Division (Abdulkadiroğlu, Sönmez, and Ünver 2004). The 3-Dimensional Stable Matching considers how to allocate $3n$ participants into n groups with each groups exactly 3 participants based on the preference over the combinations of two other participants. It is a natural extension of stable matching problem to high dimensional graph. Recently, this variant has attracted much attention (Ng and Hirschberg 1991; Boros et al. 2004; Eriksson, Sjöstrand, and Strimling 2006; Huang 2007). Many stability notions have been proposed and studied, but all notions are NP-complete.

The second problem Room Assignment-Rent Division problem concerns how to allocate rooms and share the rent among a group of friends to appeal all of them. The paper (Abdulkadiroğlu, Sönmez, and Ünver 2004) proposed an auction mechanism which mimics the market mechanism. Their model is a classic assignment game and their auction method ends up with envy free assignment-rent division. In contrast to our model, their model does not consider the influence of roommate and this is similar to notion of room envy free in our model.

2 Roommate Assignment: Market Model and Solution Concepts

The roommate market problem considers how to assign $2n$ people to n rooms with 2 people in each room. Let $I = \{1, 2, \ldots, 2n\}$ be the set of participants and $R = \{r_1, r_2, \ldots, r_n\}$ be the set of rooms. An instance includes a pair of matrices $\langle H, V\rangle$:

- $H = \{h_{ij} \mid i, j \in I, i \neq j\}$, where h_{ij} is the happiness of person i when living with j as roommates.
- $V = \{v_{ir} \mid i \in I, r \in R\}$, where v_{ir} is the valuation of person i to room r.

All happiness values h_{ij} and room evaluations v_{ir} are non-negative real numbers.

An *assignment* μ is a set of n triples (i, j, r), specifying that person i and person j live in room r. Each person appears in exactly one tuple, so does each room. We sometimes also call a perfect matching among the $2n$ people a *roommate matching*. A *rent vector* $p \in \mathbb{R}_+^{I \cup R}$ specifies the amount of rent p_i that each person $i \in I$ should pay and also the room rent p_r for each room r. If $(i, j, r) \in \mu$, then it should hold that $p_i + p_j = p_r$. In roommate market problem, we try to find a solution consisting of a pair (μ, p) of assignment and rent vector. Sometimes the rent vector is given as part of input. In that case, a solution is merely an assignment μ.

Solution Concept	Existence	Find one	Find optimal
maximum social welfare	always	NP-hard	$\frac{2}{3}$ approx. in $O(n^3)$ time
2-person stable	not always	NP-hard	N/A
4-person stable	always	$O(n^2)$	$\frac{2}{3}$-approx. in time $O(n^5)$
Person envy-free	not always	NP-hard	N/A
Room envy-free	always	$O(n^4)$	$\frac{2}{3}$-approx. in $O(n^4)$ time
Room stable	always	$O(n^3)$	$\frac{2}{3}$-approx. in $O(n^3)$ time

Table 1: The computational complexity of different stable solutions. The column "Find one" is to find one solution satisfying the property specified by the row assuming it exists. The column "Find optimal" is to find one solution maximizing social welfare among all solutions satisfying the property specified by the row.

Given μ and p, the *utility* of person i is defined as $u_i = v_{ir}+h_{ij}-p_i$, where r is the room i lives in and the roommate is j, i.e. $(i,j,r) \in \mu$. For an assignment μ, the *social welfare* is $W(\mu) = \sum_{i\in I}(u_i + p_i) = \sum_{(i,j,r)\in\mu}(h_{ij} + h_{ji} + v_{ir} + v_{jr})$. One target is to find solution with large social welfare.

Fairness is another important target other than large social welfare. Next we define some solution concepts about stability and envy-freeness. We start to consider stability for two people living in different rooms. If person i swaps with person j, then i needs to pay p_j, the rent that j is supposed to pay, and j needs to pay p_i. The next definition is a counterpart of the "exchange stability" in traditional stable roommate problem (with preference lists) (Alcalde 1994; Cechlárová and Manlove 2005; Irving 2004).

Definition 1. *A solution is* 2-person stable*, or 2PS for short, if no pair (i, j) of people living in different rooms can swap and increase both their utilities. That is, if (i, i') are assigned in room r and (j, j') are assigned in room $s \neq r$, then $v_{is} + h_{ij'} - p_j \leq v_{ir} + h_{ii'} - p_i$ or $v_{jr} + h_{ji'} - p_i \leq v_{js} + h_{jj'} - p_j$.*

One can also take the roommates of i and j into consideration, since their objection may also make the swap of i and j infeasible.

Definition 2. *A solution is* 4-person stable*, or 4PS for short, if no pair (i, j) of people living in different rooms can swap and make all 4 people in the two rooms increase their utilities.*

Since all 4 people increase their utilities implies 2 of them increase their utilities, the 4-person stability is a weaker solution concept than the 2-person stability.

We then consider *envy-freeness*, a solution concept stronger than stability. The stability basically says that no two participants prefer each other. The envy-freeness requires that no single participant prefers any other one.

Definition 3. *A solution is* person envy-free *(PEF for short) if no person i envies another person j in a different room. That is, if i is assigned in room r, paying price p_i and living with i', and j is assigned in room $s \neq r$, paying price p_j and living with j', then $u_i = v_{ir} + h_{ii'} - p_i \geq v_{is} + h_{ij'} - p_j$.*

However, unlike in one-to-one assignment games where an envy-freeness assignment always exists and can be found efficiently, it turns out that in our two-to-one setting, this person envy-freeness is way too strong to exist. It actually implies 4PS, 2PS and REF (defined next). Therefore we consider the scenario that the two people living in a room as a whole and need to move at the same time (for instance, when they are couples). This leads to the following room envy-freeness concept.

Definition 4. *A solution is* room envy-free *(REF for short) if no pair of roommates (i, j) envies any other pair of roommates (k, l). That is, if (i, j) are assigned in room r, paying price p_r in total, and (k, l) are assigned in room $s \neq r$, paying price p_s in total, then $v_{ir} + v_{jr} - p_r \geq v_{is} + v_{js} - p_s$.*

Finally, the room envy-freeness clearly implies room stability (RS), which requires that no pair of roommates (i, j) living in room r and pair of roommates (k, l) living in room s like to switch rooms so that (i, j) live in s and (k, l) live in r.

In general, we will use the term *blocking pair* to refer to the two participants or triples that break the stable or envy free condition.

The following graph shows the relationship between these concepts.

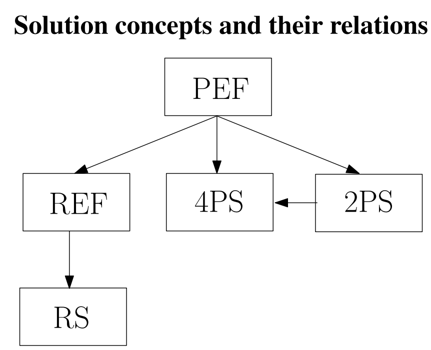

Figure 1: The arrow $A \to B$ means that concept A implies concept B.

3 Social welfare maximization

In this section, we consider how to maximize social welfare. Note that social welfare depends solely on the assignment but not on rent vector. Thus whether the rent vector is given as part of input or is required as part of output does not matter for maximizing social welfare. We will first prove the NP-hardness of finding a maximum social welfare assignment, and then give an algorithm that can find a solution with social welfare at least $\frac{2}{3}$ of the optimal in $O(n^3)$ time.

Finding maximum social welfare is NP-hard

In this section, we show that it is NP-hard to find a maximum social welfare assignment by a reduction from Tripartite Triangles Partition (TTP) problem. Given an undirected tripartite graph $G = (X \cup Y \cup Z, E)$ with $|X| = |Y| = |Z| = n$ (and X, Y, Z mutually disjoint), the Tripartite Triangle Partition problem asks to decide whether G contains n vertex-disjoint triangles. That is, whether $X \cup Y \cup Z$ has a partition $X \cup Y \cup Z = V_1 \cup V_2 \cup \cdots \cup V_n$ such that each V_i is the vertex set of a triangle in G and different V_i's are vertex-disjoint. This problem is known to be NP-complete[2]. Using this, we can prove that finding a maximum social welfare solution is also NP-hard.

Theorem 1. *It is NP-hard to find a maximum social welfare assignment, even when all happiness and evaluations take $\{0,1\}$ values.*

Proof. Given an instance $G = (X \cup Y \cup Z, E)$ of TTP, we construct an instance of market roommate problem as follows. Let $f : X \cup Y \to I$ be a one-to-one mapping that maps each vertex of $X \cup Y$ to a person, and let $d : Z \to R$ be a one-to-one mapping that maps each vertex of Z to a room. We construct an instance $\langle H, V \rangle$ as follows:

- $h_{f(x_i),f(y_j)} = h_{f(y_j),f(x_i)} = 1$ if $(x_i, y_j) \in E$;
- $v_{f(x_i),d(z_r)} = 1$ if $(x_i, z_r) \in E$;
- $v_{f(y_j),d(z_r)} = 1$ if $(y_j, z_r) \in E$;
- All other elements of H and V are 0.

It is easy to see that, for any (i, j, r), we have $h_{ij} + h_{ji} + v_{ir} + v_{jr} \leq 4$. Equality holds if and only if the corresponding vertex x, y and z form a triangle in G. Thus, let μ be a maximum social welfare assignment, then we have $SW(\mu) = 4n$ if and only if there is a perfect triangle partition of G. Therefore, we can decide whether the graph G has a perfect triangle partition by finding a maximum social welfare solution and checking whether the value is $4n$. $\square$

Approximation of maximum social welfare

Theorem 1 tells us that it is difficult to get exactly solve the maximum social welfare problem. In this section, we give a polynomial-time approximation algorithm that can find an assignment with social welfare at least $2/3$ of the optimal.

Given a set I of $2n$ people, a set R of n room, and the happiness and valuation values $\langle H, V \rangle$, we can define an undirected weighted graph G with the vertex set $I \cup R$ and edge set specified as follows. For any two distinct participants $i, j \in I$, there is an edge (i, j) with the weight $h_{ij} + h_{ji}$. For each participant $i \in I$ and each room $r \in R$, there is an edge (i, r) with the weight v_{ir}. Denote $E_H = \{(i, j) \mid i, j \in I, i \neq j\}$ and $E_V = \{(i, r) \mid i \in I, r \in R\}$. Then finding a maximum social welfare assignment is equivalent to a finding a maximum-weight triangle partition of G such that each triangle comprises one edge in E_H and two edges in E_V.

The main idea of our algorithm is to first find a maximum-weight perfect matching M_1 in subgraph $G_1 = (I, E_H)$ and a maximum-weight perfect *1-2 matching* in subgraph $G_2 = (I \cup R, E_V)$. Here a 1-2 matching of G_2 is bipartite subgraph of G_2 such that the degree of each vertex $i \in I$ is exactly 1 and the degree of each vertex $r \in R$ is exactly 2. A perfect matching in G_1 is a matching of n edges, and a perfect 1-2 matching in G_2 is a 1-2 matching of $2n$ edges. If we just take the one of M_1 and M_2, whichever has the bigger weight, to form an assignment, then we can get a (1/2)-approximation. Here we can improve this to (2/3)-approximation by combining M_1 and M_2 in a more non-trivial way.

A property is needed for understanding the description of the algorithm.

Lemma 1. *Given a graph $G = (I \cup R, E_H \cup E_V)$, let M_1 be a perfect matching in $G_1 = (I, E_H)$ and M_2 a perfect 1-2 matching in $G_2 = (I \cup R, E_V)$. Then the following two properties hold.*

1. *The subgraph $M_1 \cup M_2$ is a vertex-disjoint cycle cover of G. That is, $M_1 \cup M_2$ is a collection of vertex-disjoint cycles and every vertex in G is covered by one cycle.*
2. *For any cycle in $M_1 \cup M_2$, every 3 consecutive vertices consist of one vertex of R and two vertices of I.*

Proof. The first property can be obtained by the following observation: the degree of each vertex in subgraph $M_1 \cup M_2$ is exactly 2. Thus the subgraph is a collection of vertex-disjoint cycles.

The second property follows from the fact that in the subgraph $M_1 \cup M_2$, each vertex of R connects to two vertics of I and each vertex of I connects to one vertex of I and one vertex of R. If we start from a vertex of R, then it must follows two vertices of I. If start from a vertex of I, then it follows either a vertex of I and a vertex of R or a vertex of R and a vertex of I. In any case, every 3 consecutive vertices consist of one vertex of R and two vertices of I. $\square$

With these two properties, we give the following Double-Matching algorithm.

Now we prove the following theorem.

Theorem 2. *Double-Matching algorithm outputs an assignment with weight at least $2/3$ of the optimal in $O(n^3)$ time.*

Proof. We first show that the algorithm can guarantee $2/3$ approximation ratio. Given a matching M, let $w(M)$ denote the total weight of that matching. It is easy to see that $w(M_1) + w(M_2)$ is at least as big as the maximum social welfare, because for any optimal solution, the sum of happiness is no greater than $w(M_1)$ and the sum of valuations is no greater than $w(M_2)$.

Next, we removed some edges, but note that the total weight of the removed edges is at most $(w(M_1) + w(M_2))/3$, since on each cycle (of length at least 2), the removed weight is at most 1/3 of the weight of that cycle. Thus, the assignment has social welfare at least $\frac{2}{3}(w(M_1) +$

[2]Rizzi demonstrated the proof in lecture note "NP-Complete Problem: Partition into Triangles"; see http://profs.sci.univr.it/~rizzi/classes/Complexity/provette/Mirko/pt_fine.pdf. Basically, it is proven by a reduction from *3-dimensional matching*, one of Karp's 21 NP-complete problems (Garey and Johnson 1979).

Algorithm 1 Double-Matching

Find a maximum-weight perfect matching M_1 of $G_1 = (I, E_H)$.
Find a maximum-weight perfect perfect 1-2-matching M_2 in bipartite graph $G_2 = (I, R, E_V)$.
Let $C = \{C_1, C_2 \dots, C_k\}$ the set of cycles of $M_1 \cup M_2$.
for $i = 0$ to k **do**
  Denote the edges of C_i by $e_1, \dots, e_{3l}$ in a cyclic order (starting from an arbitrary vertex).
  if $l \geq 2$ **then**
    for $t = 0$ to 2 **do**
      $W_t = \sum_{j=1,\dots,3l:\ j \equiv t \bmod 3} w(e_j)$.
    end for
    Let $t^* \in \{0, 1, 2\}$ be a minimizer of $\min_t W_t$.
    Remove edges e_j for all $j \equiv t^* \bmod 3$.
  end if
end for
The subgraph is now a collection of vertex-disjoint connected triples (i, j, r).
return The set of these triples (i, j, r) as an assignment.

$w(M_2))$, which is in turn at least $2/3$ of the maximum social welfare.

Now we analyze the time complexity of the algorithm. The problem of finding a maximum weighted matching in general graph can be solved in $O(|V||E| + |V|^2 \log |V|)$ time (Gabow 1990). In our case, $|E_H| = \Theta(n^2)$, thus it takes time $O(n^3)$ to find M_1. The second matching M_2 can be found in $O(n^3)$ time either by a b-matching algorithm ($b = 2$ in our case) (Rajabi-Alni, Bagheri, and Minaei-Bidgoli 2014; Kleinschmidt and Schannath 1995), or the following trick: first make a copy r' for each room vertex r, then set $v_{ir'} = v_{ir}$. It is easy to see that a maximum-weight matching M_2' in this extended graph G' corresponds to the maximum-weight 1-2 matching M_2 of the original graph G, with $w(M_2') = w(M_2)$. Indeed, given a perfect matching M_2' on G', we can get a perfect 1-2 matching M_2 in G by assigning the neighbors of r and r' in M_2' both to r in G.

After finding M_1 and M_2, the remaining part of the Double-Matching algorithm can be easily implemented in $O(n^2)$ time. Combining all the cost, the algorithm runs in at most $O(n^3)$ time. □

4 Person stable solutions

In this section, we consider various person stable solution concepts. The notion of person stability is an extension of exchange stability in traditional stable roommate model (Alcalde 1994; Cechlárová and Manlove 2005; Irving 2004), in which a matching is exchange stable if no pair of people living in different rooms like to switch their positions. In our model, we add the valuation v_{ir} of rooms and quantify the preference lists by happiness value h_{ij}. In this section, we assume that the room rent $\{p_r\}$ is given as part of input. We first show that finding a 2-person stable solution is NP-hard, and then we demonstrate an algorithm that can find a 4-person stable solution efficiently. Note that the algorithm works for any given room rent $\{p_r\}$, which makes it widely applicable.

2-Person stable

Our definition of 2-person stable is reminiscent of the exchange stable of Stable Roommate (SR) problem. Formally, in a given instance of SR, each of $2n$ people ranks the others in strict order of preference. A matching M is *exchange stable* if there are no two participants i and j, each of whom prefers the other's partner to her own partner in M. It is NP-complete to decide whether an input instance I admits an exchange stable solution (Cechlárová and Manlove 2005). We next use this to show the hardness of finding a 2-person stable solution in our roommate market model. Note that search problem is in general harder than decision problem: If one can efficiently find a desirable solution assuming it exists, then one can use this algorithm to decide whether such solution exists by first running the algorithm and then verify the outputted solution. The next theorem shows that even the (easier) decision problem is NP-hard.

Theorem 3. *Deciding whether a given input instance (H, V, p) of roommate market admits a 2 person stable solution is NP-hard.*

Proof. We will reduce the exchange stable roommate problem to this one. Given $2n$ preference lists as an instance of stable roommate problem of $2n$ people, we construct an instance for our roommate market as follows: if j is the k-th most preferred person in person i's list, then let $h_{ij} = 2n-k$. Set all room evaluations $v_{ir} = 0$ and all rents $p_i = c$ for some constant c.

Since the all room evaluations are 0 and rent for all people are the same, the utility of each person depends solely on the happiness h_{ij} of the roommate j. It is easy to see that a solution is 2PS in our model if and only if matching is exchange stable in the given instance of roommate stable problem. Thus deciding the existence of 2PS solution in our model is at least as hard as deciding whether the existence of exchange stable problem in standard roommate problem, which is NP-hard. □

4-Person stable

In contrast to the above nonexistence result, next we will show that a 4PS solution always exists for any given input (H, V, p), and we can find one in $O(n^2)$ time. Furthermore, in $O(n^5)$ time we can find a 4PS solution with the social welfare at least $2/3$ of the optimal (over all solutions).

First we design a greedy algorithm following the *serial dictatorship mechanism* (Abdulkadiroğlu and Sönmez 1998) to find a 4PS solution efficiently. We call a pair (i, j) of people living different rooms a *4PS blocking pair* if swapping them makes them and their original roommate all increase their individual utilities. Then a 4PS solution is just one without any 4PS blocking pair.

Theorem 4. *For any given input (H, V) and the room rent $\{p_r\}$, we can find a 4PS solution in $O(n^2)$ time.*

Proof. First we assign person 1 to the room r_1 which maximizes $v_{1r_1} - p_{r_1}/2$, together with person j_1 which maximizes h_{1j_1}. Let person 1 and person j_1 each pay $p_{r_1}/2$.

Then we remove people 1 and j_1 from the list of people, and remove r_1 from the list of rooms. Among the remaining people, pick the person i_2 with the smallest index, and put her in the room r_2 which maximizes $v_{i_2 r_2} - p_{r_2}/2$ among the available rooms, and assign to her person j_2 which maximizes $h_{i_2 j_2}$ among the available people as her roommate. Again let person i_2 and j_2 share the cost p_{r_2} evenly. Continue this process until no one is left.

The algorithm has n rounds and each round needs to scan two lists of length at most $2n$, so the total time is $O(n^2)$. Next we analyze the correctness. No 4PS blocking pair can involve person 1 since she already gets her most preferred room and roommate. (Here involving person 1 means that either person 1 or her roommate is in the blocking pair.) Put the triple $(1, j_1, r_1)$ aside and apply the same argument, one can see that no 4PS blocking pair can involve i_2 or j_2. Continuing this argument to the very end rules out all possible blocking pairs. Thus the algorithm obtains a 4-person stable solution. □

Next we put social welfare into consideration. We first give the following "Algorithm 2: Local Search".

Algorithm 2 Local Search

Start from assignment outputted by Double-Matching algorithm.
while there is a 4PS blocking pair (i, j) **do**
  Swap the people i and j.
end while
return The current room assignment, and the two roommates pay the room rent evenly.

Theorem 5. *For any given (H, V) and room rent $\{p_r\}$'s, the Local Search algorithm outputs a 4PS solution with social welfare at least $2/3$ of the optimal in $O(n^5)$ time.*

Proof. By definition of 4PS, when swapping i and j in each iteration of the **while** loop, no person's utility drops, thus the social welfare is always increasing. Since we start from an assignment with social welfare at least 2/3 of the optimal, the social welfare at the end is at least this much.

The algorithm clearly outputs a 4PS solution since otherwise by definition there is a 4PS blocking pair, and thus the **while** loop has not ended yet. Next we will show that the algorithm ends within $O(n^3)$ iterations. This implies the desirable time bound: Since each iteration we only need to check over all pairs (i, j) for a 4PS block pair, the overall time is $O(n^3) \cdot O(n^2) = O(n^5)$.

Now we bound the number of iterations of the **while** loop. Note that in each iteration, the swapping "produces" two triples (i_1, j_1, r_1) and (i_2, j_2, r_2), where i_b and j_b live in r_b $(b = 1, 2)$ after the swapping. We claim that each triple (i, j, r) can be "produced" only once. Indeed, notice that the utilities of i and j increase after the swapping (of i or j to someone in some other room). But as long as i and j live in room r, the sum of their utilities is a fixed constant $v_{ir} + v_{jr} + h_{ij} + h_{ji} - p_r$. Thus, any triple (i, j, r) can be produced only once in the swapping. As the number of the triples (i, j, r) is $O(n^3)$ and each iteration produces 2 such triples, the **while** loop ends in at most $O(n^3)$ iterations. □

(From the proof we can also see that actually even if we strengthen the definition of 4PS to that one of the blocking pair i and j strictly increase her utility, and the other three people involved do not decrease their individual utilities, the algorithm still finds such a solution. We only need to adapt the proof by defining a swap "producing" a triple (i, j, r) if the total utility of person i and j increases by the swap.)

5 Room envy-freeness, stability and price

In this section, we focus on the room envy-free solutions. We will show that given any roommate assignment, a room envy-free solution can be found efficiently. Indeed, after the roommate assignment is fixed, we consider the two roommates living in one room as a whole "agent", then apply the results of *assignment game* (Shapley and Shubik 1971) in a two-sided market to get an assignment and a price, such that the matching between the n agents and n rooms is envy-free.

Formally, given an roommate assignment β (which is a perfect matching among the $2n$ people), the algorithm treats each $(i, j) \in \beta$ as an agent a_k. The valuation of a_k to each room r is defined as $w_{kr} = v_{ir} + v_{jr}$. For this new two-sided market with n agents and n rooms, there always exists a room price p_r for each room r and a perfect matching between agents and rooms such that if a_k matches to r, then $w_{kr} - p_r \geq w_{kr'} - p_{r'}$ for any other room r' (Shapley and Shubik 1971). Such a solution can be found in $O(n^4)$ time by Hungarian Method (Dütting, Henzinger, and Weber 2013).

Since the definition of room envy-freeness treats the two roommates as a whole, the above solution is also a room envy-free solution for our model. Furthermore, one can make the solution to have a large social welfare by first using Double Matching algorithm to find an assignment, and then use assignment game to further improve the social welfare.

Theorem 6. *Given any roommate assignment input (H, V), a room envy-free solution (μ, p) with social welfare at least 2/3 of the optimal can be found in $O(n^4)$ time.*

Finally, recall that a solution is room stable if no pair of roommates (i, j) living in room r and pair of roommates (k, l) living in room s like to switch rooms so that (i, j) live in s and (k, l) live in r. Such solutions can be easily found in time $O(n^3)$. Indeed, for any pair-up $\{(i_1, j_1), ..., (i_n, j_n)\}$ of people, find a maximum matching among these n pairs and the n rooms, with weight between (i_k, j_k) and r_ℓ set to be $h_{i_k j_k} + h_{j_k i_k} + v_{i_k r_\ell} + v_{j_k r_\ell}$. Then it is easily seen that the solution is room stable. If we use the pair-up as in a 2/3 approximation algorithm, then the solution achieves at least 2/3 of the optimal social welfare.

Acknowledgments

We thank Xiaohui Bei, Heng Zhang and Jiacheng Zhuo for some interesting discussions. The work was partially supported by Research Grants Council of the Hong Kong S.A.R. (Project no. CUHK419413).

References

Abdulkadiroğlu, A., and Sönmez, T. 1998. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica* 689–701.

Abdulkadiroğlu, A.; Sönmez, T.; and Ünver, M. U. 2004. Room assignment-rent division: A market approach. *Social Choice and Welfare* 22(3):515–538.

Alcalde, J. 1994. Exchange-proofness or divorce-proofness? stability in one-sided matching markets. *Economic Design* 1(1):275–287.

Baıou, M., and Balinski, M. 2000. Many-to-many matching: stable polyandrous polygamy (or polygamous polyandry). *Discrete Applied Mathematics* 101(1):1–12.

Bansal, V.; Agrawal, A.; and Malhotra, V. S. 2003. Stable marriages with multiple partners: efficient search for an optimal solution. In *Automata, Languages and Programming*. Springer. 527–542.

Boros, E.; Gurvich, V.; Jaslar, S.; and Krasner, D. 2004. Stable matchings in three-sided systems with cyclic preferences. *Discrete Mathematics* 289(1):1–10.

Cechlárová, K., and Manlove, D. F. 2005. The exchange-stable marriage problem. *Discrete Applied Mathematics* 152(1):109–122.

Dütting, P.; Henzinger, M.; and Weber, I. 2013. Sponsored search, market equilibria, and the hungarian method. *Information Processing Letters* 113(3):67–73.

Eriksson, K.; Sjöstrand, J.; and Strimling, P. 2006. Three-dimensional stable matching with cyclic preferences. *Mathematical Social Sciences* 52(1):77–87.

Gabow, H. N. 1990. *Data structures for weighted matching and nearest common ancestors with linking*. University of Colorado, Boulder, Department of Computer Science.

Gale, D., and Shapley, L. S. 1962. College admissions and the stability of marriage. *American mathematical monthly* 9–15.

Garey, M., and Johnson, D. 1979. Computers and intractability: A guide to the theory of np-completeness.

Huang, C.-C. 2007. Twos company, threesa crowd: Stable family and threesome roommates problems. In *Algorithms–ESA 2007*. Springer. 558–569.

Irving, R. W.; Manlove, D. F.; and Scott, S. 2000. The hospitals/residents problem with ties. In *Algorithm Theory-SWAT 2000*. Springer. 259–271.

Irving, R. W.; Manlove, D. F.; and Scott, S. 2003. Strong stability in the hospitals/residents problem. In *STACS 2003*. Springer. 439–450.

Irving, R. W. 1985. An efficient algorithm for the stable roommates problem. *Journal of Algorithms* 6(4):577–595.

Irving, R. W. 2004. The man-exchange stable marriage problem. *Department of Computing Science, Research Report, TR-2004-177, University of Glasgow, UK.*

Kleinschmidt, P., and Schannath, H. 1995. A strongly polynomial algorithm for the transportation problem. *Mathematical Programming* 68(1-3):1–13.

Knuth, D. E. 1997. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*, volume 10. American Mathematical Soc.

Malhotra, V. S. 2004. On the stability of multiple partner stable marriages with ties. In *Algorithms–ESA 2004*. Springer. 508–519.

Ng, C., and Hirschberg, D. S. 1991. Three-dimensional stabl matching problems. *SIAM Journal on Discrete Mathematics* 4(2):245–252.

Rajabi-Alni, F.; Bagheri, A.; and Minaei-Bidgoli, B. 2014. An $O(n^3)$ time algorithm for the maximum weight b-matching problem on bipartite graphs. *arXiv preprint arXiv:1410.3408*.

Roth, A. E., and Sotomayor, M. A. O. 1992. *Two-sided matching: A study in game-theoretic modeling and analysis*. Number 18. Cambridge University Press.

Roth, A. E.; Sönmez, T.; and Ünver, M. U. 2005. Pairwise kidney exchange. *Journal of Economic theory* 125(2):151–188.

Shapley, L. S., and Shubik, M. 1971. The assignment game i: The core. *International Journal of game theory* 1(1):111–130.