Online Roommate Allocation Problem*

Guangda Huzhang[1], Xin Huang[2], Shengyu Zhang[3], Xiaohui Bei[4]
[1,4]School of Physical and Mathematical Sciences, Nanyang Technological University
[2,3]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[1]ghuzhang001@e.ntu.edu.sg, [2,3]{xhuang,syzhang}@cse.cuhk.edu.hk, [4]xhbei@ntu.edu.sg

Abstract

We study the online allocation problem under a roommate market model introduced in [Chan *et al.*, 2016]. Consider a fixed supply of n rooms and a list of $2n$ applicants arriving online in random order. The problem is to assign a room to each person upon her arrival, such that after the algorithm terminates, each room is shared by exactly two people. We focus on two objectives: (1) maximizing the social welfare, which is defined as the sum of valuations that applicants have for their rooms, plus the happiness value between each pair of roommates; (2) satisfying the stability property that no small group of people would be willing to switch roommates or rooms.

We first show a polynomial-time online algorithm that achieves a constant competitive ratio for social welfare maximization. We then extend it to the case where each room is assigned to $c > 2$ people, and achieve a competitive ratio of $\Omega(1/c^2)$. Finally, we show both positive and negative results in satisfying various stability conditions in this online setting.

1 Introduction

Online allocation studies the problem in which input information is revealed step by step, and the algorithm is required to make irrevocable decisions in each step without the knowledge of future input items. Data come in an online fashion for certain resource allocation problems. For example, in the Google AdWords problem, keywords arrive sequentially in real time, and after observing a keyword query, Google needs to decide immediately and irrevocably what advertisement to display to maximize its revenue.

In this paper, we study the problem of online resource allocation in a roommate assignment setting proposed in [Chan *et al.*, 2016]. The objective in a roommate market problem is to match rooms to applicants under certain budget constraints. In public massive housing program, applicants are required to fill in a form to state their preference over rooms and roommates. The model also applies to applications such as university accommodation and conference roommate arrangement.

Formally, in the online roommate market model, there are n rooms, and also $2n$ persons that arrive online in random order (later we will generalize this to a general model with c-bed rooms). Each person has a valuation for each room and a happiness valuation of each potential roommate. An allocation is an assignment of each person to some room, such that each room contains exactly two persons. The utility of a person is defined as the sum of his/her happiness for roommate and valuation for the room.

There are several dimensions to measure the quality of an allocation. From a global perspective, a natural objective is to maximize the allocation's *social welfare*, which is the summation of utilities of all persons. However, such efficiency may come at the cost of unfairness, with some individuals allocated very little resource. In this regard, we focus on the *stability* of an allocation. An allocation is stable if no coalition of a small number of people can devise new trades to make everyone in the coalition better off.

Our goal is to design online algorithms on roommate market model that perform well in these two measures. Our main results are summarized as follows.

(a) We give a polynomial-time online algorithm with a constant competitive ratio with respect to the optimal social welfare.

(b) For the generalized c-bed model for any $c \geq 2$, we extend our algorithm and obtain a $\Omega(1/c^2)$ competitive ratio.

(c) We show both positive and negative results on whether various stability conditions can be achieved in the online model.

1.1 Related Works

Our model follows the work by Chan et al. [2016] on the roommate market. In their work, the authors focused on the offline 2-bed problem and presented constant approximation algorithms for social welfare maximization. They also proposed various solution concepts on stability and envy-freeness, and studied their existence and the computational complexity of the corresponding search problems. The biggest difference between our works is that we focus on the online version of the problem and also generalize the results to the c-bed setting.

*This work was supported by Research Grants Council of the Hong Kong S.A.R. (Project no. CUHK14239416)

Besides the work of Chan et al. [2016], there are two lines of work closely related to this paper: table matching and online bipartite matching. The stable matching problem has a long history dating back to 1960s [Gale and Shapley, 1962]. After decades of development the area grows into a rich theory with a vast body of literature; see surveys [Knuth, 1997; Iwama and Miyazaki, 2008]. The basic stable matching problem has been extended in many dimensions. One direction is to consider general matchings [Irving, 1985; Irving and Manlove, 2002] and higher dimensional matchings [Ng and Hirschberg, 1991; Eriksson *et al.*, 2006; Huang, 2007]. Different stability notions have also been proposed, such as exchange stable [Cechlárová and Manlove, 2005] and popular matching [Biró *et al.*, 2010].

The study of online bipartite matching is originated from the secretary problem, which asks the best strategy to choose one secretary from n candidates coming in an online fashion with random arriving order. The optimal algorithm is $\frac{1}{e}$-competitive in expectation; see [Ferguson, 1989] for historical detail. The matroid generalization was introduced in the work [Babaioff *et al.*, 2007], and inspired a line of works [Dimitrov and Plaxton, 2008; Im and Wang, 2011; Soto, 2013; Gharan and Vondrák, 2013]. The optimal competitive ratio of unweighted online bipartite matching with adversarial arrival order was proved to be $1 - \frac{1}{e}$ [Karp *et al.*, 1990]. Later Kessel et al. [2013] extended the idea to weighted bipartite matching with random arrival order, and obtained optimal competitive ratio $\frac{1}{e}$ for this model. Many variants were proposed and analyzed, such as vertex-weighted matching [Aggarwal *et al.*, 2011] and online packing [Kesselheim *et al.*, 2014].

2 Preliminary

We are given a set of $2n$ agents $I = \{1, 2, ..., 2n\}$, a set of n rooms $R = \{r_1, r_2, ..., r_n\}$, a happiness matrix $H = \{h_{ij} \mid i, j \in I, i \neq j\}$ in which h_{ij} denotes the happiness of agent i when she is assigned to live with agent j, and a valuation matrix $V = \{v_{ir} \mid i \in I, r \in R\}$ in which v_{ir} denotes the valuation of agent i to room j. We assume that all happiness and valuation values are nonnegative. The outcome of the roommate market is an allocation $A = \{(i, j, r)\}$ which consists of n disjoint triple. Triple (i, j, r) means agent i and j are assigned together to room r. We require that every agent is assigned to one room and every room is assigned to exactly 2 agents.

The social welfare of an allocation A is defined as $SW(A) = \sum_{(i,j,r)\in A}(h_{ij} + h_{ji} + v_{ir} + v_{jr})$.

We assume an online setting where all agents arrive online in uniformly random order. When agent i arrives, her valuation v_{ir} to every room r, as well as her happiness value h_{ij} to all agents j that have already arrived, are revealed to the algorithm, and the algorithm needs to assign agent i to some available room immediately. Note that since there are exactly $2n$ agents and n rooms, we are not allowed to leave any agent unassigned. Our goal is to find an allocation A that can maximize $\mathbb{E}[SW(A)]$, where the expectation is taken over both the randomness of the algorithm and the random arriving order of the agents.

An online algorithm is said to be *c-competitive* (or to have *competitive ratio* c), if its output allocation has expected social welfare no less than $c \cdot SW(A_{opt})$, where A_{opt} is the optimal offline allocation.

3 Online Roommate Market

In this section, we present an online roommate market allocation algorithm that achieves constant competitive ratio.

3.1 Online No-Rejection Bipartite Matching

Our algorithm is built on an online bipartite matching algorithm with a no-rejection condition. Recall that in the standard online bipartite matching setting, we are given a weighted complete bipartite graph $G = (L, R, E)$ with $|L| = |R| = n$. The vertices in R are given in advance. The vertices in L arrive online in a random order and the edges incident to each vertex $l \in L$ are revealed when l arrives. An algorithm should either assign the current vertex to an unmatched adjacent vertex in R immediately, or leave it unassigned. The objective is usually to maximize the total weight of the resulting matching.

This online bipartite matching problem has been studied extensively in the literature [Aggarwal *et al.*, 2011; Kalyanasundaram and Pruhs, 1993; Kesselheim *et al.*, 2014]. However, the problem we consider here has an important difference: we do not allow the algorithm to leave any agent unassigned. Such no-rejection feature brings new challenges to the online algorithm design. Next we present our algorithm, which can be viewed as an extension of the online bipartite algorithm proposed in [Kesselheim *et al.*, 2013].

Algorithm 1 ONLINEMATCHING(n, R)

$counter \leftarrow 0$
$L \leftarrow \emptyset$
$A \leftarrow \emptyset$
for every person v comes **do**
    $L \leftarrow L \cup \{v\}$
    $counter \leftarrow counter + 1$
    if $counter \geq n/5$ **then**
        $M^v \leftarrow$ Optimal matching on $G[L \cup R]$
        $e^v \leftarrow$ The matching edge that contains v in M^v
        if $A \cup e^v$ is a matching **then**
            $A \leftarrow A \cup e^v$
        else
            Randomly choose an available vertex v'.
            $A \leftarrow A \cup (v, v')$
    else
        Randomly choose an available vertex v'.
        $A \leftarrow A \cup (v, v')$
return A

Lemma 1. ONLINEMATCHING *is a polynomial time and c_b-competitive algorithm for the online no-rejection bipartite graph matching problem, where* $c_b = \frac{\ln 5 - 0.8}{5} \approx 0.1618$.

The proof is similar to that of Algorithm 1 in [Kesselheim *et al.*, 2013] and is omitted here due to space constraint.

3.2 Constant Approximation Algorithm for Online Roommate Market

We now present our constant competitive ratio algorithm for the online roommate market problem. It uses the online no-rejection bipartite matching algorithm as a key ingredient. The high level idea is that we first apply Algorithm ONLINEMATCHING on the first n people arrived. After this stage, each room contains exactly one agent. Then we combine each room-person pair as one new "room", and apply Algorithm ONLINEMATCHING again on the last n people with adjusted valuations to match them to the n room-person pairs.

Algorithm 2 ONLINEROOMMATE (n, H, V)

1: Run ONLINEMATCHING on the first n agents arrived.
2: Let M_1 be the output matching.
3: **for** every agent i arrived after the first n agents **do**
4:     **for** each room $r \in R$ **do**
5:         Set $v'_{ir} \leftarrow v_{ir} + (h_{ij} + h_{ji})$ where $(j, r) \in M_1$
6: Run ONLINEMATCHING on the last n agents with valuation matrix V'.
7: Let M_2 be the returned matching.
8: **return** $M_1 \cup M_2$

Theorem 1. *Algorithm* ONLINEROOMMATE *is a polynomial time and $\frac{c_b}{4}$-competitive algorithm for the online roommate market problem, where $c_b = \frac{\ln 5 - 0.8}{5} \approx 0.1618$.*

Proof. Let A_{opt} denote the optimal offline allocation with maximum social welfare. Let M_{pp} denote the maximum weight general graph matching between the $2n$ agents, where the weight between agent i and j is $h_{ij} + h_{ji}$. Let M_{pr} denote the maximum weight matching between $2n$ agents and n rooms where each room is duplicated into 2 vertices. By slight abuse of notations, in the following we use $SW(M)$ to denote the summation of the edge weights in matching M.

The social welfare $SW(A_{opt})$ can be divided to two parts: the first part is the happiness between roommates, which will not exceed $SW(M_{pp})$; the other part is the valuations between agents and the rooms, which will not exceed $SW(M_{pr})$. Hence we have

$$SW(M_{pp}) + SW(M_{pr}) \geq SW(A_{opt}).$$

Next we bound $SW(M_{pp})$ and $SW(M_{pr})$. Fix a particular agents arriving order. Let A_1 be the set of first n agents, A_2 be the set of last n agents, and E_{12} be the set of weighted edges between A_1 and A_2 (where again the weight of edge (i, j) between agent i and agent j is $h_{ij} + h_{ji}$). We further define the following notations:

- M_{pb}: the maximum weight matching in bipartite graph (A_1, A_2, E_{12}).
- M_{pr1}: the maximum weight matching between the first n agents and n rooms
- M_{pr2}: the maximum weight matching between the last n agents and n rooms.

We will show in the following that

$$\begin{aligned}&2\mathbb{E}[SW(M_{pr1}) + SW(M_{pr2}) + SW(M_{pb})]\\ \geq& SW(M_{pp}) + SW(M_{pr})\end{aligned}$$

where the expection is over the random arriving order of the agents.

First we bound $SW(M_{pb})$. Since agents arrives in uniformly random order, every edge in M_{pp} will be present in E_{12} with probability at least $\frac{1}{2}$, and these edges together form a matching. We therefore have

$$\mathbb{E}[SW(M_{pb})] \geq \sum_{(i,j)\in M_{pp}} \frac{1}{2}(h_{ij} + h_{ji}) = \frac{1}{2}SW(M_{pp})$$

Now we bound $SW(M_{pr})$ by $SW(M_{pr1})$ and $SW(M_{pr2})$. Let M_{pr0} be the maximum weight bipartite matching between $2n$ people and n rooms and each room only has one slot. We have

$$SW(M_{pr1}) + SW(M_{pr2}) \geq SW(M_{pr0}) \geq \frac{1}{2}SW(M_{pr}).$$

The first inequality is because the edges in $M_{pr1} \cup M_{pr2}$ can at least cover all edges in M_{pr0}. Together with

$$\mathbb{E}[SW(M_{pb})] \geq \frac{1}{2}SW(M_{pp}),$$

we have

$$\begin{aligned}&2\mathbb{E}[SW(M_{pr1}) + SW(M_{pr2}) + SW(M_{pb})]\\ \geq& SW(M_{pp}) + SW(M_{pr})\\ \geq& SW(A_{opt}).\end{aligned}$$

Back to our algorithm, the first call to ONLINEMATCHING gives us a matching with expected social welfare no less than $c_b \cdot SW(M_{pr1})$; the second call to ONLINEMATCHING gives us a matching with expected social welfare no less than $c_b \cdot \max\{SW(M_{pb}), SW(M_{pr2})\}$. Let A denote the allocation output by our algorithm. Together we have

$$\begin{aligned}&\mathbb{E}[SW(A)]\\ \geq& c_b \cdot \mathbb{E}[SW(M_{pr1})] + c_b \cdot \max\{\mathbb{E}[SW(M_{pb})], \mathbb{E}[SW(M_{pr2})]\}\\ \geq& c_b \cdot \mathbb{E}[SW(M_{pr1})] + \frac{c_b}{2}\mathbb{E}[SW(M_{pb}) + SW(M_{pr2})]\\ \geq& \frac{c_b}{2}\mathbb{E}[SW(M_{pr1}) + SW(M_{pb}) + SW(M_{pr2})]\\ \geq& \frac{c_b}{4}SW(A_{opt}).\end{aligned}$$

□

The above results can also be generalized to the case where the number of agents is not exactly $2n$. Note that in this case, we need to adjust the model to either allow that some room contains less than 2 people (when the number of agents is smaller than $2n$), or that some agent is left unassigned (when the number of agents is more than $2n$).

Corollary 2. *There is a constant competitive ratio algorithm for online roommate market problem with n rooms and the number of agents $p = O(n)$.*

The proof details are omitted due to space constraints.

3.3 Online Roommate Market Problem with Unknown Number of Agents

Note that a prerequisite for Corollary 2 to hold is that the algorithm must know the number of agents p beforehand. Perhaps surprisingly, this also turns out to be a necessary requirement. In this section we show that if the number of agents p is unknown to the algorithm, then no online algorithm that can achieve constant competitive ratio for the online bipartite matching problem as well as the online roommate market problem.

Lemma 3. *If the number of the agents is unknown in an online bipartite matching problem, no online algorithm can achieve a constant competitive ratio.*

Due to space constraints, we only present a sketch of the proof. Given any constant $\epsilon > 0$, we prove the lemma by constructing a class of bipartite graphs such that no algorithm can achieve competitive ratio ϵ for all graphs.

We restrict ourself to the following type of bipartite graphs: there exists only one vertex $r^* \in R$ that has nonzero valuations for the online vertices, i.e., $v_{ir} = 0$ for all $i \in L$ and $r \neq r^* \in R$. We can represent such a graph by a *value multiset* $\{v_{1r^*}, v_{2r^*}, \ldots, v_{nr^*}\}$. Thus the decision that the algorithm needs to make is to select upon arrival which vertex in L to match to r^*, and the optimal offline solution is always $\max_i v_{ir^*}$.

Next we construct graph sets $B_0, \ldots, B_m$ for any value of m, where each B_k is a class of bipartite graphs that satisfy the following properties:

1. Every graph in B_k has the same number of online vertices, denoted by b_k, and their value multiset is supported on the set $\{1, L, L^2, \ldots, L^k\}$, where L is a large enough constant (say at least $\frac{2}{\epsilon}$). This implies that when given a graph in B_k as the input, in order to guarantee ϵ-competitive ratio, the algorithm needs to select an optimal online vertex (i.e., vertex with value L^k) with probability at least $\frac{\epsilon}{2}$.
2. $b_0 = 1$ and $b_{k+1} > b_k$ for all $k \geq 0$. Given any graph $G_{k+1} \in B_{k+1}$ as the input, after seeing its b_k random vertices, with high probability the values seen coincide with the value multiset of some graph in B_k. Therefore the algorithm will not be able to distinguish whether this input is from G_{k+1} or from some graph in B_k.

The construction details are omitted. The technique is similar to the construction of a particular exponentially distributed probability distribution used in other contexts such as [Hajiaghayi *et al.*, 2007; Gharan and Vondrák, 2013].

From these properties we know that to be ϵ competitive on any input graph $G_m \in B_m$, the algorithm must match some vertex to r^* in time steps $[b_{m-1}+1, b_m]$ with probability at least $\frac{\epsilon}{2}$. By an induction argument, we can show that the algorithm must match some vertex to r^* with probability at least $\frac{\epsilon}{2}$ from each time step interval $[b_k+1, \ldots, b_{k+1}]$ for all $0 \leq k < m$. However, when we set $m > \frac{2}{\epsilon}$, the sum of these probabilities will be greater than 1, and the task becomes impossible because the algorithm can match at most one online vertex to r^*.

Note that an algorithm for the online roommate market problem with constant competitive ratio would imply a constant competitive ratio algorithm for the online bipartite matching problem. Thus Lemma 3 implies that the former also cannot exist.

Corollary 4. *If the number of the agents is unknown in an online roommate market problem, there is no online algorithm that can achieve a constant competitive ratio.*

4 Generalized c-Bed Model

Our algorithm can also be generalized to the case where each room can take more than 2 people. In a *generalized online roommate market problem*, there are cn people and n rooms, and we want to assign c people to every room, with the goal of maximizing the social welfare

$$SW(A) = \sum_{(i_1,i_2,\ldots,i_c,r)\in A} \left(\sum_{1\leq j<k\leq c} h_{i_j i_k} + \sum_{j=1}^{c} v_{i_j r} \right)$$

The other settings are the same as the standard roommate market problem.

Similar to the original case, we divide the cn agents into c blocks with n agents in every block. Then we apply ONLINEMATCHING c times, each time assign a block of n agents to n rooms using an updated valuation matrix.

Algorithm 3 ONLINECBEDROOMMATE (n, H, V)

1: $V' \leftarrow V$
2: **for** $g = 1, 2, \ldots, c$ **do**
3: Run ONLINEMATCHING on the next n arriving agents with valuation matrix V'.
4: Let M_g be the returned matching.
5: **for** every agent i that is yet to arrive **do**
6: $v'_{ir} \leftarrow v'_{ir} + h_{ij} + h_{ji}$ where $(j, r) \in M_g$
7: **return** $\cup M_g$

Theorem 2. ONLINECBEDROOMMATE *has competitive ratio* $\frac{c_b(c-1)}{c^3}$ *for the generalized online roommate market problem, where* $c_b = \frac{\ln 5 - 0.8}{5} \approx 0.1618$.

Proof. The proof is along similar lines of that for Theorem 1.

Let A_{opt} denote the optimal offline allocation with maximum social welfare. Let A_{pp} denote the allocation of cn agents into n rooms that with maximum total happiness value between agents, and let M_{pr} denote the maximum bipartite matching between cn agents and n rooms where each each room is duplicated into c copies. Again it is easy to see that $SW(A_{opt}) \leq SW(A_{pp}) + SW(M_{pr})$.

Note that for each room r, the c agents in this room will have total happiness contributed by $\frac{c(c-1)}{2}$ pairs of relations. Let C_r be the general graph *matching* with maximum total happiness among the c agents allocated to room r in allocation A_{pp}. We can show that $SW(A^r_{pp}) \leq c \cdot SW(C_r)$, where A^r_{pp} is the sum of happiness of all agents in room r in allocation A_{pp}. This is because each clique of c vertices can always be covered by c matchings via the round-robin tournament algorithm.

Now go back to the online random arrival order. We call every n consecutively arriving vertices a block. The probability of any two vertices arrive in different blocks is $\frac{cn-n}{cn-1}$. Let M_{pp}^{ij} represent the maximum weight bipartite matching between agents in the ith block and jth block, and we can assume $c > 2$. We can distribute all matching edges in each C_r into some M_{pp}^{ij} if the two vertices of the edge are in different blocks, and every vertex will be involved in at most one matching edge. Hence by linearity we have

$$\begin{aligned}\sum_{1\le i<j\le c} \mathbb{E}[SW(M_{pp}^{ij})] &\ge \frac{cn-n}{cn-1}\sum_{r=1}^{n} SW(C_r)\\ &> \frac{c-1}{c}\sum_{r=1}^{n} SW(C_r)\end{aligned}$$

Put everything together, we have

$$\begin{aligned}SW(A_{pp}) = \sum_{r=1}^{n} SW(A_{pp}^{r}) &\le \sum_{r=1}^{n} c\cdot SW(C_r)\\ &\le \sum_{1\le i<j\le c} \frac{c^2}{c-1}\cdot \mathbb{E}[SW(M_{pp}^{ij})]\end{aligned}$$

Next we look at M_{pr}. Let M_{pr}^{1} be the matching induced from M_{pr} such that each room connects with only one agent who has the largest matching value. We have $SW(M_{pr}) \le c\cdot SW(M_{pr}^{1})$. Let M_{ir} be the maximum weight matching between agents in i-th block and one slot of each room. We also have $SW(M_{pr}^{1}) \le \sum_{i=1}^{c} SW(M_{ir})$. Combing these two inequalities gives us

$$SW(M_{pr}) \le c\sum_{i=1}^{c} SW(M_{ir}) < \frac{c^2}{c-1}\sum_{i=1}^{c} SW(M_{ir}).$$

Together with A_{pp}, we have

$$\begin{aligned}&SW(A_{opt}) \le SW(A_{pp}) + SW(M_{pr})\\ &\le \frac{c^2}{c-1}\cdot \mathbb{E}\left[\sum_{1\le i<j\le c} SW(M_{pp}^{ij}) + \sum_{i=1}^{c} SW(M_{ir})\right]\end{aligned}$$

Finally, let M_i be the maximum weight matching between the agents in i-th block and aggregated agent-room combinations according to the algorithm. We have

$$SW(M_i) \ge \max\{SW(M_{ir}), \max_{1\le j\le i-1} SW(M_{pp}^{ij})\}.$$

Since there are at most c items in the max bracket, we have $c\cdot SW(M_i) \ge SW(M_{ir}) + \sum_{j=1}^{i-1} SW(M_{pp}^{ij})$. Thus,

$$\begin{aligned}&\frac{c^2}{c-1}\cdot \mathbb{E}\left[\sum_{i=1}^{c} c\cdot SW(M_i)\right]\\ &\ge \frac{c^2}{c-1}\cdot \mathbb{E}\left[\sum_{1\le i<j\le c} SW(M_{pp}^{ij}) + \sum_{i=1}^{c} SW(M_{ir})\right]\\ &\ge SW(A_{pp}) + SW(M_{pr}) \ge SW(A_{opt})\end{aligned}$$

By Lemma 1, the algorithm computes M_i^* which satisfies $\mathbb{E}[SW(M_i^*)] \ge c_b \cdot SW(M_i)$. Therefore, we have

$$\mathbb{E}\left[\sum_{i=1}^{c} SW(M_i^*)\right] \ge \frac{c_b(c-1)}{c^3} SW(A_{opt}).$$

This finishes the proof of this theorem. □

4.1 Rooms with Different Capacities

The model can be further generalized to allow rooms to have different capacities. Assume we have n agents and k rooms with capacities $\vec{c} = (c_1, c_2, \ldots, c_k)$ such that $\sum_i c_i = n$.

Algorithm 4 ONLINEGENCBEDROOMMATE $(\vec{c}, H, V)$

$c \leftarrow \max\{c_1, \ldots, c_k\}$
$V' \leftarrow V$
for $g = 1, 2, \ldots, c$ **do**
  Let n_g be the number of available rooms.
  Run ONLINEMATCHING on the next n_g arriving agents with valuation matrix V' (only rooms with open capacities are used).
  Let M_g be the returned matching.
  for every agent i that is yet to arrive **do**
    $v'_{ir} \leftarrow v'_{ir} + h_{ij} + h_{ji}$ where $(j, r) \in M_g$
  for every room i with $c_i > 0$ **do**
    $c_i \leftarrow c_i - 1$
return $\cup M_i$

Corollary 5. ONLINEGENCBEDROOMMATE *achieves constant competitive ratio for the generalized online roommate market problem where every room has constant capacity.*

The proof is similar to that for Theorem 2 and is omitted.

5 Stability Results

In this section we discuss different stability conditions in the online roommate market model.

We mainly focus on the stable notions introduced by Chan et al. [2016]. Note that it is always more difficult to achieve stability conditions in the online setting than in the offline setting. Thus we will only discuss stability notions that guarantee to exist in the offline setting. These notions are the *4-person stability* and *room stability*. In addition, we will also consider a new stable notion – *weak room stability*.

Our goal is to design online algorithms that can satisfy certain stability conditions together with strong social welfare guarantees. Note that this is not always achievable for all stability notions we mention above. Our results can be summarized in the following table.

Stability type	Achievable	Social welfare competitive ratio
4-person stable	no	-
Room stable	yes	unknown
Weakly room stable	yes	constant

5.1 4-Person Stable

Definition 1. *[Chan* et al.*, 2016] An allocation is* 4-person stable *if for any agents i and agent j in two different rooms, swapping them cannot make all 4 people in these two rooms strictly increase their utility.*

In the offline setting, [Chan *et al.*, 2016] gave an algorithm that can find a 4-person stable solution in $O(n^2)$ time. However, in the following we show that in the online setting, no algorithm can always guarantee a 4-person stable solution.

Lemma 6. *No algorithm can always find a 4-person stable allocation in the online roommate market setting.*

Proof. We prove this lemma by a simple example with 4 agents and 2 rooms. Assume every agent has valuation 0 for every room. Hence we only need to consider the happiness values between them. We also assume happiness values are symmetric, i.e., $h_{ij} = h_{ji}$ for every agent i and j. Consider the following agents arrival sequence: let 1 and 2 be the first and second arriving agents with $h_{12} = 1$. When agent 2 arrives, any algorithm needs to make one of two choices:

- Assign agent 1 and 2 to the same room. In this case, assume that the next two arriving agents 3 and 4 have happiness values $h_{13} = h_{24} = 100$. All other unspecified happiness values are 0. It is easy to check that this already breaks the 4-person stable condition because swapping 1 and 3 would make every agent better off.
- Assign agent 1 and 2 to different rooms. In this case, assume that the next two arriving agents 3 and 4 have $h_{34} = 1$ and all other happiness values are 0. Here moving agent 1 and 2 to the same room can improve the utility of every agent.

Note that an online algorithm need to make an assignment decision at each moment some agent arrives. This means regardless of what this algorithm does, there always exists a problem instance and a particular agents arriving, such that the algorithm does not output a 4-person stable solution. □

5.2 Room Stability and Weak Room Stability

Recall the definition of room stability as follows.

Definition 2. *An allocation is* room stable *if for any two agents i, i' in room r_i and two agents j, j' in another room r_j, switching their rooms cannot increase the sum of the two roommates' utilities for both rooms.*

When discussing this condition, the happiness value between roommates can be ignored because the roommate relation will not be changed. It turns out that this room stable condition can be satisfied by an online algorithm.

Lemma 7. *There is an online algorithm that always gives a room stable allocation.*

Proof. The simple serial dictatorship algorithm works as following: For every arriving agent, assign this agent to his/her most preferred room.

Now we show that the simple dictatorship algorithm that assigns every arriving agent her most preferred available room can produce a room stable allocation. Fixing any two rooms r_1 and r_2, let A, B, C, D denote the 4 agents assigned to these two rooms by this algorithm. Suppose the arriving order among them is A, B, C, D. If A and B both choose the same room, then they would not want to move to the other room. If A and B choose different rooms, without loss of generality, assume A chooses room r_1 and B chooses room r_2. If C chooses room r_1, then A and C both prefer room r_1 to r_2; If C chooses r_2, both B and C prefer room r_2 to r_1. In either case, there is a room in which the two tenants do not want to switch. □

We comment that the above dictatorship algorithm, while always preserving the room stability, does not have any competitive ratio guarantees on social welfare. It remains an open question to design an algorithm that can achieve both room stability and constant competitive ratio on social welfare. However, as we will show below, if we are willing to weaken the room stablility condition, such goal indeed becomes achievable.

Definition 3. *An allocation is* weakly room stable *if for any two agents i, i' in room r_i and two agents j, j' in another room r_j, switching their rooms cannot increase all four agents' utilities.*

Theorem 3. *There is an online algorithm that can always produce a weakly room stable allocation with competitive ratio $c_b/8$ on social welfare, where $c_b = \frac{\ln 5 - 0.8}{5} \approx 0.1618$.*

Proof. Recall in the proof of Theorem 1, we showed

$$2\mathbb{E}[SW(M_{pr1}) + SW(M_{pb}) + SW(M_{pr2})] \geq SW(A_{opt}).$$

Note that we also have $\mathbb{E}[SW(M_{pr1})] = \mathbb{E}[SW(M_{pr2})]$. This means we can ignore one of these two terms and still get a constant competitive ratio solution. Thus we modify algorithm ONLINEROOMMATE as follows: for each of the first n arriving agents, we just choose the best empty room available to her. For the next n agent we still follow algorithm ONLINEMATCHING. After this change, our new algorithm will have competitive ratio $c_b/8$. In addition, the output solution also satisfies weak room stability. This is because if we want to swap room r_i and r_j, and first slot of r_i is assign before r_j. Then the agent who is assigned to r_i will not want to switch because she (weakly) prefers room r_i to r_j. Thus the output allocation is always weakly room stable. □

6 Conclusion

This paper studies an online version of the roommate problem with stochastic arrivals, proposes a constant-factor competitive algorithm, and generalizes the results to the case of different number of agents per room. It also shows both positive and negative results in satisfying different stability conditions in this online setting. This model aims to capture a general online resource allocation scenario in which a resource can be assigned to multiple agents, and agents sharing the same resource exhibit positive externalities. That is, the valuation of an agent to a resource also depends on the identity of other agents who are assigned to the same resource.

The framework leaves a number of future working directions. For example, it remains open how close to optimal the provided algorithms are. It is also worth taking strategic behavior into consideration and study truthful mechanisms in the online roommate problem setting.

References

[Aggarwal *et al.*, 2011] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1253–1264. SIAM, 2011.

[Babaioff *et al.*, 2007] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 434–443. Society for Industrial and Applied Mathematics, 2007.

[Biró *et al.*, 2010] Péter Biró, Robert W Irving, and David F Manlove. Popular matchings in the marriage and roommates problems. In *Algorithms and Complexity*, pages 97–108. Springer, 2010.

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

[Chan *et al.*, 2016] Pak Hay Chan, Xin Huang, Zhengyang Liu, Chihao Zhang, and Shengyu Zhang. Assignment and pricing in roommate market. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 446–452, 2016.

[Dimitrov and Plaxton, 2008] Nedialko B Dimitrov and C Greg Plaxton. Competitive weighted matching in transversal matroids. In *International Colloquium on Automata, Languages, and Programming*, pages 397–408. Springer, 2008.

[Eriksson *et al.*, 2006] Kimmo Eriksson, Jonas Sjöstrand, and Pontus Strimling. Three-dimensional stable matching with cyclic preferences. *Mathematical Social Sciences*, 52(1):77–87, 2006.

[Ferguson, 1989] Thomas S Ferguson. Who solved the secretary problem? *Statistical science*, pages 282–289, 1989.

[Gale and Shapley, 1962] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *American mathematical monthly*, pages 9–15, 1962.

[Gharan and Vondrák, 2013] Shayan Oveis Gharan and Jan Vondrák. On variants of the matroid secretary problem. *Algorithmica*, 67(4):472–497, 2013.

[Hajiaghayi *et al.*, 2007] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *Thirty-Frist AAAI Conference on Artificial Intelligence*, volume 7, pages 58–65, 2007.

[Huang, 2007] Chien-Chung Huang. Two's company, three'sa crowd: Stable family and threesome roommates problems. In *Algorithms–ESA 2007*, pages 558–569. Springer, 2007.

[Im and Wang, 2011] Sungjin Im and Yajun Wang. Secretary problems: Laminar matroid and interval scheduling. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1265–1274. SIAM, 2011.

[Irving and Manlove, 2002] Robert W Irving and David F Manlove. The stable roommates problem with ties. *Journal of Algorithms*, 43(1):85–105, 2002.

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

[Iwama and Miyazaki, 2008] Kazuo Iwama and Shuichi Miyazaki. A survey of the stable marriage problem and its variants. In *Informatics Education and Research for Knowledge-Circulating Society, 2008. ICKS 2008. International Conference on*, pages 131–136. IEEE, 2008.

[Kalyanasundaram and Pruhs, 1993] Bala Kalyanasundaram and Kirk Pruhs. Online weighted matching. *Journal of Algorithms*, 14(3):478–488, 1993.

[Karp *et al.*, 1990] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[Kesselheim *et al.*, 2013] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Algorithms–ESA 2013*, pages 589–600. Springer, 2013.

[Kesselheim *et al.*, 2014] Thomas Kesselheim, Andreas Tönnis, Klaus Radke, and Berthold Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 303–312. ACM, 2014.

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

[Ng and Hirschberg, 1991] Cheng Ng and Daniel S Hirschberg. Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics*, 4(2):245–252, 1991.

[Soto, 2013] José A Soto. Matroid secretary problem in the random-assignment model. *SIAM Journal on Computing*, 42(1):178–211, 2013.