CSE-4111, Artificial Intelligent Lab

Mirajul Mohin, Roll: FH-28

April 22, 2019

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

Problem Definition

This research introduces an online algorithm for roommate market model. Let, a roommate market model contains n fixed rooms and 2n number of agents. Now the problem is to assign a room for each person upon his/her arrival and thus after termination each room contains exactly two persons. Two things need to be satisfied (1) maximizing social welfare (2) satisfying stability property. It first addresses an online algorithm (where each room contains exactly two person) which works in polynomial time having a constant competitive ratio. And then extends this for the case where each room can contain c > 2 people.

Method

Here, given a set of n rooms $R = \{r_1, r_2, r_3, ..., r_{\mathrm{n}}\}$, a set of 2n agents $I = \{1, 2, 3, ...2n\}$, a happiness matrix $H = \{h_{\mathrm{ij}} | i, j \epsilon I, i \neq j\}$, a valuation matrix $V = \{v_{\mathrm{ir}} | i \epsilon I, r \epsilon R\}$ and the outcome of this model is $A = \{(i, j, r)\}$, which means that agent i is assigned to agent j in room r. Social welfare of an allocation A is $SW(A) = \sum_{(i,j,r)\epsilon A}(h_{\mathrm{ij}} + h_{\mathrm{ji}} + v_{\mathrm{ir}} + v_{\mathrm{jr}})$

Here it is assumed an online setting where each agent arrive in a uniform random order and immediately after arrival agents need to be assigned to a room.

For the first part, total number of agents is $2n$ where total number of rooms is n. So, each room will contain exactly two people. For the second part, total number of agents is cn, where n is the total room number. So, each room can contain c number of people.

The main goal is to find an allocation that maximizes social welfare $SW(A)$, where the expectation is taken over both the randomness of the algorithm and the random arriving order of the agents.

Input/Output (Part-1)

$$V = \begin{bmatrix} 5 & 2 \\ 7 & 9 \\ 2 & 1 \\ 7 & 9 \end{bmatrix}$$

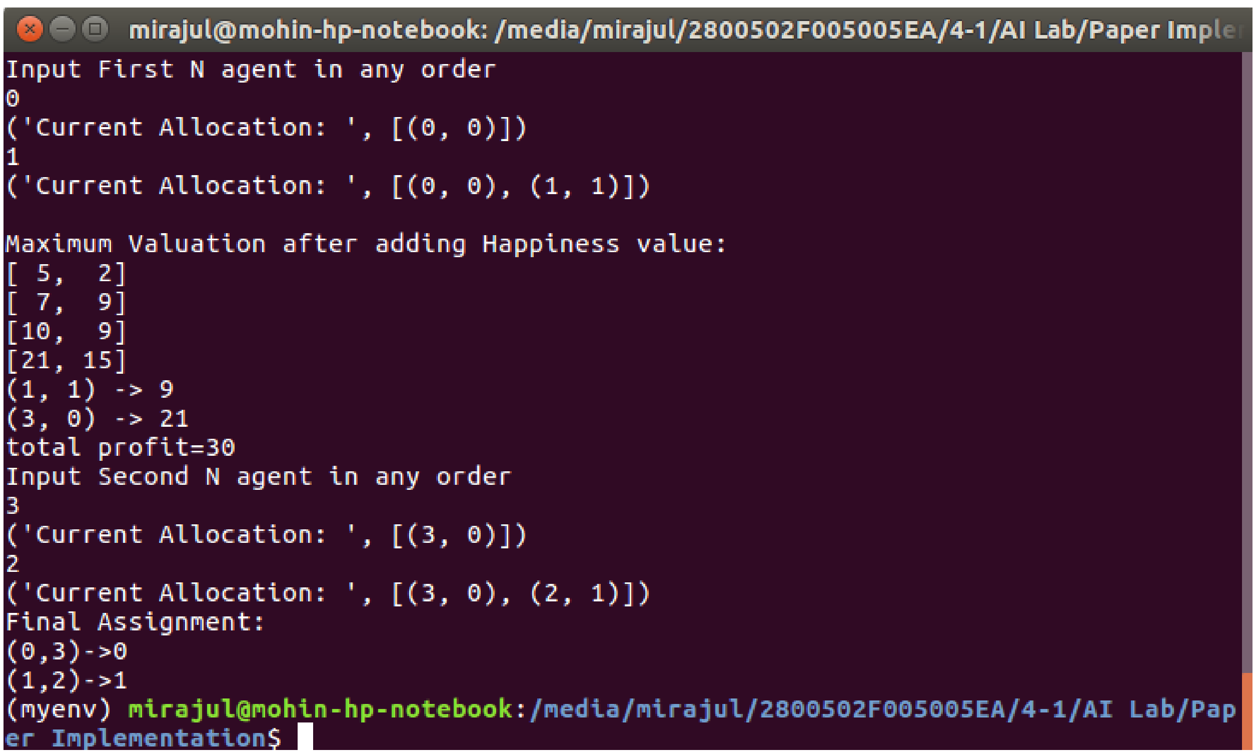
Valuation matrix V

$$H = \begin{bmatrix} 0 & 5 & 3 & 7 \\ 2 & 0 & 5 & 3 \\ 5 & 3 & 0 & 1 \\ 7 & 3 & 2 & 0 \end{bmatrix}$$

Happiness matrix H

For n = 2, total number of rooms is 2, total number of agents is 4. So the dimension of Valuation matrix is 4 x 2 (2n x n) and the dimension of Happiness matrix is 4 x 4 (2n x 2n).

If four agents come following 0, 1, 3, 2 this sequence, for two matrix mentioned above my implementation gives two allocation ([0, 3], 0) and ([1, 2], 1). That means agent-0, agent-3 will be in room-0 and agent-1, agent-2 will be in room-1.



```
mirajul@mohin-hp-notebook: /media/mirajul/2800502F005005EA/4-1/AI Lab/Paper Imple
Input First N agent in any order
0
('Current Allocation: ', [(0, 0)])
1
('Current Allocation: ', [(0, 0), (1, 1)])

Maximum Valuation after adding Happiness value:
[ 5,  2]
[ 7,  9]
[10,  9]
[21, 15]
(1, 1) -> 9
(3, 0) -> 21
total profit=30
Input Second N agent in any order
3
('Current Allocation: ', [(3, 0)])
2
('Current Allocation: ', [(3, 0), (2, 1)])
Final Assignment:
(0,3)->0
(1,2)->1
(myenv) mirajul@mohin-hp-notebook:/media/mirajul/2800502F005005EA/4-1/AI Lab/Pap
er Implementation$
```

Figure 1: Figure showing allocation for above input

The above figure shows an allocation for 4 agents, 2 rooms and for the above Valuation and Happiness matrix.

Input/Output (Part-2)

This part generalizes the previous online market problem where each room can take c people, where c > 2. Here, total number of agent is cn, number of rooms is n. So, each room will contain c number of people. It just uses the previous algorithm to run c times for updated V matrix.

$$V = \begin{bmatrix} 5 & 2 \\ 7 & 9 \\ 2 & 1 \\ 7 & 9 \\ 6 & 2 \\ 5 & 3 \end{bmatrix} \qquad H = \begin{bmatrix} 0 & 5 & 3 & 7 & 2 & 9 \\ 2 & 0 & 5 & 3 & 6 & 1 \\ 5 & 3 & 0 & 1 & 4 & 8 \\ 7 & 3 & 2 & 0 & 3 & 5 \\ 2 & 5 & 3 & 9 & 0 & 7 \\ 3 & 2 & 1 & 5 & 9 & 0 \end{bmatrix}$$

Valuation matrix V Happiness matrix H

For n = 2 and c = 3 , total number of rooms is 2, total number of agents is 6. So the dimension of Valuation matrix is 6 x 2 (cn x n) and the dimension of Happiness matrix is 6 x 6 (cn x cn).

If six agents come following 0, 1, 3, 2, 5, 4 this sequence, for two matrix mentioned above my implementation gives two allocation ([0, 3, 4], 0) and ([1, 2, 5], 1). That means agent-0, agent-3 and agent-4 will be in room-0 and agent-1, agent-1, agent-2 and agent-5 will be in room-1.

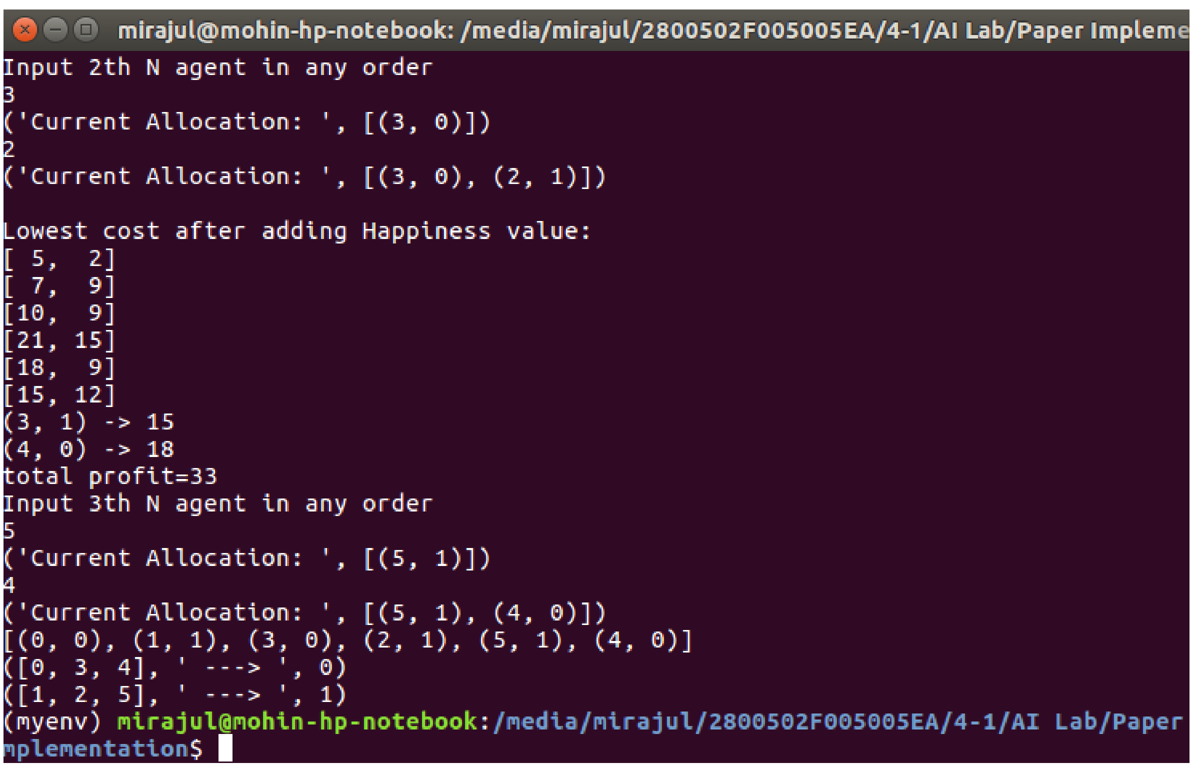


Figure 2: Figure showing allocation for above input

Discussion

Three algorithms, ONLINEMATCHING(n, R), ONLINEROOMMATE (n, H, V) and ONLINECBEDROOMMATE (n, H, V) are used. Here ONLINEMATCHING(n, R) is a polynomial time algorithm having c_b the competitive ratio, where $c_\mathrm{b} = \frac{\ln 5 - 0.8}{5} \approx 0.1618$ *[Kesselheim et al., 2013]*

They have proposed ONLINEROOMMATE (n, H, V) which is a polynomial time constant $\frac{c_\mathrm{b}}{4}$-competitive algorithm [$c_\mathrm{b} = \frac{\ln 5 - 0.8}{5} \approx 0.1618$]. Constant competitive ratio has been calculated comparing with the same off-line allocation.

The last one, ONLINECBEDROOMMATE (n, H, V), is the extended version of proposed ONLINEROOMMATE (n, H, V) algorithm. This algorithm has competitive ratio $\frac{c_\mathrm{b}(c-1)}{c^3}$ [$c_\mathrm{b} = \frac{\ln 5 - 0.8}{5} \approx 0.1618$].

The next part of this research addresses another algorithm where rooms are with different capacities. Besides, they have proved some stability conditions and some other things related to it.