Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Dormitory Assignment Using a Genetic Algorithm

Chih-Ching Chang & Che-Chern Lin

To cite this article: Chih-Ching Chang & Che-Chern Lin (2021) Dormitory Assignment Using a Genetic Algorithm, Applied Artificial Intelligence, 35:15, 2276-2297, DOI: 10.1080/08839514.2021.1999595

To link to this article: https://doi.org/10.1080/08839514.2021.1999595

Published online: 08 Jan 2022.

Submit your article to this journal

Article views: 1094

View related articles

CrossMark
View Crossmark data

APPLIED ARTIFICIAL INTELLIGENCE
2021, VOL. 35, NO. 15, 2276–2297
https://doi.org/10.1080/08839514.2021.1999595

Dormitory Assignment Using a Genetic Algorithm

Chih-Ching Chang and Che-Chern Lin

Department of Software Engineering and Management, National Kaohsiung Normal University, Kaohsiung City, Taiwan

ABSTRACT

This study proposes a genetic algorithm based algorithm for assigning freshmen's dormitory rooms according to five living habits and preferences. In the proposed genetic algorithm, we used a locally exhaustive crossover method to avoid divergence results, and then got better fitness values for this dormitory assignment problem. In addition, we used a half-half selection strategy to reduce the time consumption during the iteration procedure. Experimental results have shown that the proposed algorithm could have acceptable performances with reasonable computational time. In addition, two counterpart methods were used to evaluate the performance of the proposed algorithm: a simulated annealing method and a random assignment method. The comparative results have also shown that 1) the execution time of the proposed algorithm was significantly less that of the simulated annealing method; 2) the fitness value of the proposed algorithm was significantly less than that of the random assignment method; 3) the fitness value of the proposed algorithm is almost the same as that of the simulated annealing method; 4) the proposed algorithm is stable to repeated executions; 5) the proposed algorithm is still suitable even when the capacities (beds) of rooms are different.

ARTICLE HISTORY
Received 9 February 2021
Revised 16 October 2021
Accepted 20 October 2021

Introduction

Dormitory assignment is one of important issues in school administration. Each year when freshmen come to universities, schools' dormitory management staffs need to arrange dormitory rooms for the freshmen. Traditional dormitory assignment methods might specify dormitory rooms by a random method or simply by students' identification numbers. The traditional methods do not consider students' living habits and preferences such that some students might request to change their rooms due to the diversities of the

CONTACT Che-Chern Lin ✉ cclin@nknu.edu.tw Department of Software Engineering and Management, National Kaohsiung Normal University, 62, Shenjhong Rd., Yanchao District, Kaohsiung City 82444, Taiwan

Chih-Ching Chang received a B.S. from I-Shou University, Taiwan, and a M.S. from National Kaohsiung Normal University, Taiwan. He is currently working at National Kaohsiung Normal University. His research interest includes genetic algorithms, information systems, and artificial intelligence.Che-Chern Lin received a B.S. in EE from National Taiwan University of Science and Technology, Taiwan, in 1985, a M.S. in EE from National Taiwan University, Taiwan, in 1987, and a Ph.D. in EE from University of Pittsburgh, USA, in 1996. He is currently a professor in the Department of Software Engineering and Management at National Kaohsiung Normal University, Kaohsiung, Taiwan. His research interest includes artificial intelligence, fuzzy systems, genetic algorithms, object-oriented systems, and e-learning systems.

habits and preferences between them and their roommates. Based on (Chang 2019), this study proposes a more intelligent method to assign students' dormitory rooms with the consideration of students' habits and preferences.

How to match students' habits and preferences is originally a stable marriage problem (SMP), and was first proposed by Gale and Shapley in 1962 (Gale and Shapley 1962). It is basically a matching problem to find optimal matching between men and women according to preference lists of opposite gender. Mathematically, the SMP can be formulized as follows (Iwama and Miyazaki 2008). Consider an instant set of I consisting of n men and n women. If a man m prefers woman 1 (denoted as w_1) to woman 2 (denoted as w_2), we use notation $w_1 \ >_m w_2$ to express the about preference order in m's preference list of women. Similarly, $m_1 \ >_w m_2$ expresses the preference order that a women w prefers men 1 (m_1) to man 2 (m_2) in her preference list of men. A match set M of I is a set that consists of a series of disjoint man-woman matching pairs of I. Consider a man m and a woman w forming a matching pair. We use notations $M(m) = w$ and $M(w) = m$ to indicate this matching pair of w and m. A block pair for M is formed for m and w, if the following three conditions are met (Iwama and Miyazaki 2008).

(i) M(m) ≠ w;

(ii) $w >_m M(m)$;

(iii) $m >_w M(w)$.

M is unstable if there is a block pair for M, otherwise, M is stable. A SMP is to find stable matching for I. Besides, Gale and Shapley have also proposed an algorithm, called the Gale-Shapley Algorithm, to find a stable matching of I with time $\boldsymbol{O}(n^2)$ (Gale and Shapley 1962).

Essentially, a SMP is an optimization problem with respect to four kinds of matching criteria (Le Hong, Hoang Huu, and Chung 2016). Consider a - man m and a woman w in I. $mr(m, w)$ is the rank of w in m's preference list, and $wr(w, m)$ is the rank of m in w's preference list. The cost functions for the SMP are defined as follows (Le Hong, Hoang Huu, and Chung 2016).

$$Cost_{man} = \sum_{(m,w)\in M} mr(m, w)$$

$$Cost_{woman} = \sum_{(m,w)\in M} wr(w, m)$$

where M is the set of matching pairs of men and women in I.

The optimization problems are based on four kinds of matching criteria as follows (Le Hong, Hoang Huu, and Chung 2016).

- Man-optimal, which minimizes $Cost_{man}$
- Woman-optimal, which minimizes $Cost_{woman}$.
- Egalitarian, which minimizes $(Cost_{man} + Cost_{woman})$
- Sex-equal, which minimizes $|Cost_{man} - Cost_{woman}|$

Originally inspired by the Darwin's Theory of Evolution, Genetic algorithms (GAs) are one of well-known artificial intelligence methods to get solutions, where genes and chromosomes are the fundamental elements to perform GA computing. Similar to the Darwin's Theory of Evolution, a chromosome is constructed by a sequence of genes. A fitness function is used to evaluate the performance of a GA according to its chromosomes. Basically, GA computing follows an evolutionally iterative process to find solutions within acceptable computational time. There are three basic operations in GAs: crossover, selection, and mutation. Crossover methods recombine the genes from the chromosomes of parent generations, and thus to rebuild new possible chromosomes for child generations. The common crossover methods include one-point, two-point, and uniform crossovers. Selection methods are utilized to choose better chromosomes from a pool of parent generations in order to produce possible child generations. Two popular selection methods are employed in GAs: roulette-wheel and tournament selections. A mutation operation randomly changes the values of genes under a certain probability which is called a mutation rate. The criteria for the SMP might be case by case with variant types of models, and the searching space for these problems might be huge. Thus, to simplify the computational complexity, this study proposes a GA to assign dormitory rooms for freshmen according to five living habits and preferences. In the proposed GA, we used a locally exhaustive crossover method to avoid divergence results, and then got better fitness values for this dormitory assignment problem. In addition, we also used a half-half selection strategy when performing a roulette-wheel selection, in order to reduce the time consumption during the GA' iteration procedure. This study also conducted experiments to validate the performance of the proposed algorithm, where the data were collected from 691 freshmen at a national university in southern Taiwan.

Related Work

A general survey about the SMP has been presented by (Iwama and Miyazaki 2008). In this survey, a basic type of SMP has been defined mathematically together with different variants. These variants included incomplete preference list, preference list with ties, incomplete list with ties, one-sided preference list, stable roommate problem, hospitals/residents problem, man-exchanging stable marriage, many-to-many stable marriage, student-project allocation problem, and 3-dimensional stable matching problem (Iwama and Miyazaki 2008). Ren et al. (2021) have generally discussed the nature of matching problems, and have classified these matching problems into two categories: explicit matching and implicit matching. The explicit matching included three models: one-to-one, many-to-one, and many-to-many (Ren et al. 2021); on the other hand, the implicit matching consisted of retrieve matching, user-item matching, entity-relation matching, and image matching (Ren et al. 2021). To solve a tie and bounded preference list problem for SMP, Trang et al. (2019) have proposed an algorithm to get a maximum cardinality weakly stable matching. The algorithm proposed by Trang et al. began with a random selection of matching pairs, and then found out the most and the least constrained variables to delete blocking pairs by an iterative procedure.

The SMP is one of the important topics in the area of operation research. Many real-world problems related to the SMP have been solved in different areas such as computer science, healthcare, education, and management, etc. For example, in computer networks, the problem of choosing suitable switches and hosts is essentially a matching problem and was solved by (Alimudin, Winarno, and Ishida 2019). In wireless communications, the matching of device-to-device (D2D) communication is similar to the SMP (Xiao et al. 2016). Socially aware D2D communication can transfer the social relationship among user devices to useful information in order to select a suitable physical D2D communication system (Xiao et al. 2016). Therefore a framework has been presented to analyze the socially aware D2D communication, as a belief-based stable marriage game, in order to solve a spectrum sharing problem (Xiao et al. 2016). In computer vision, Madi, Paquet, and Kheddouci (2019) decomposed a 3-D object into a set of sub-structures, and proposed a SMP-based algorithm to solve 3-D object recognition and classification problems.

In healthcare systems, how to determine the best matching between patients and physicians is one of the important issues. Chen, Chen, and Yang (2020) have proposed a framework to reduce patients' waiting time and to improve the matching effectiveness. In this framework, a discrete deferred acceptance scheme with the consideration of issuance types has been utilized to reduce the waiting time and regrets, instead of using a traditional continuous deferred acceptance scheme. Experimental results have shown that the framework proposed by Chen et al. could reduce the waiting time of finding suitable physicians for patients. Team-matching problems, a variant of the SMP, have been widely utilized in many areas such as education, management, government administration, etc. Meulbroek et al. (2019) have applied the Gale-Shapley Algorithm and a web-based team management tool called CATME to build student teams in order to choose the students' favorite projects according to their preference lists. Experimental results have shown that with the algorithm proposed by Meulbroek et al. 93.84% of students could select top three favorite projects according to their preference lists (Meulbroek et al. 2019). For developing countries such as Bangladesh, people might have special needs (e.g., location, rent, house space, etc.) to select their rented houses. Therefore, Bristi, Chowdhury, and Sharmin (2019) have proposed a framework for tenants to select their house owners based on the Gale-Shapley Algorithm. Experimental results have also shown that with the framework proposed by Bristi et al., tenants and house owners received high satisfactory about the tenant-owner matching results (Bristi, Chowdhury, and Sharmin 2019).

The dormitory room assignment problem is furthermore considered as a stable-roommates problem (Gale and Shapley 1962; Irving 1985; Pittel and Irving 1994), trying to find best matching pairs under certain criteria. In such a problem, if the number of persons being matched increases, the probability of finding feasible solutions decreases (Pittel and Irving 1994). The stable-roommates problem can be further extended to crew matching problem (Katarína and Ferková 2004) and kidney exchange problem (Roth, Sönmez, and Ünver 2005). Irving (1985) has presented an algorithm to efficiently examine the stable-matching problem. Practically, in stable-roommates problems, if the numbers of rooms and persons are large, the searching space is huge. Moreover, some studies have presented mathematics-based methods to solve the stable-roommates problems using theorems and proofs (Abraham, Biró, and Manlove 2005; Biró, Iñarra, and Molis 2016; Irving and Manlove 2002; Tan 1990). These mathematical methods might require huge searching spaces to solve the problems. To reduce searching spaces, some studies used soft computing methods in order to find sub-optimal solutions with acceptable computational time. These soft computing methods include simulated annealing algorithms, fuzzy systems, and evolutionary algorithms.

Some early studies used GAs to solve the SMP (Nakamura et al. 1995; Vien and Chung 2006). For, example, in the work by Nakamura et al., a sex-equal SMP has been solved where a lattice structure had been built according to the preference lists of men and women. The lattice structure was then transferred to a directed graphic to indicate the transitions on the lattice structure. A node in the directed graphic indicated a bit in a chromosome string. In the work by Vien and Chung, four criteria have been presented to evaluate matching performance: man-optimal, women-optimal, egalitarian, and sex-equal (Vien and Chung 2006); the chromosome structure of their work simply used arrays to indicate the preference lists for men and women. They have also presented a crossover method for two parent chromosomes. This crossover method followed three steps: (1) making a copy of the first parent; (2) selecting an arbitrary part from the second parent; (3) making minimum changes to fit the selected part (Vien and Chung 2006).

Recently, Albalawi and Maashi (2021) have proposed a GA-based framework to select and optimize a software development life cycle (SDLC) where two parts were utilized: a selection mechanism and an optimization process. In the selection mechanism, a selection matrix was established to indicate the relationship between project parameters (e.g., clarity of requirements, project size, complexity, etc.) and software development techniques (e.g., waterfall, agile, spiral, etc.). In the optimization process, a GA was applied to find the best SDLC where a chromosome represented a SDLC, and each gene in this chromosome indicated a single phase of this SDLC. A multi-points crossover was then employed to make reproductions. Experimental results have shown that the framework proposed by Albalawi and Maashi could reduce the completion time for different SDLC models (Albalawi and Maashi 2021). Zayed and Maashi (2021) have proposed a hybrid framework to optimize software testing where two stages were conducted: a maximum process and a minimization process. In the maximization process, a GA was used to maximize test coverage; in the minimization process, a greedy algorithm was utilized to minimize the size of test suite. In the GA proposed by Zayed and Maashi, a chromosome represented a testing case, and a gene in this chromosome indicated a statement. A single-point crossover was then conducted to make reproductions. Zayed and Maashi argued that their framework could solve real-world testing problems and consequently satisfy the software requirements of high reliability, security, and safety (Zayed and Maashi 2021).

Based on students' habits and preferences, Settanni (2000) has proposed a method to solve a dormitory room assignment problem using a simulated annealing algorithm. Moreover, Trung and Anh (2009) have proposed three modified simulated annealing algorithms for dormitory room assignment problems: informed simulated annealing (ISA), simulated annealing with non-monotonic reheating (SA_REHEAT), and very fast simulated re-annealing (VFSA). In the work by Trung and Anh, two kinds of conditions were employed: hard conditions (e.g., gender, room capacity, students' special needs, etc.) and soft conditions (e.g., types of rooms, bed locations, floor number, learning habits, music and game preferences, etc.). Trung and Anh have further indicated that ISA had a shortest computational time but its performance was worse; SA_REHEAT had a best performance with acceptable time; VFSA had unstable computation time to get acceptable results. In

Table 1. The comparative table between the experiment in (Eslami and Afzali 2007) and our experiment.

Item	Experiment in (Eslami and Afzali 2007)	Our experiment
Numbers of students	10	33, 290, and 691
Numbers of rooms	5	9, 73, and 173
Gene encoding	A gene represents a single student.	A gene represents a single student.
Chromosome	A chromosome represents a single room, which contains a one-to-one matching pair of genes.	A chromosome represents a single room, which contains multiple genes
Habits and preferences	Nationality, religion, and course	As indicated in Section 3.1.
Selection method	Tournament	As indicated in Section 3.2.
Crossover method	Mask and cyclic	As indicated in Section 3.2.
Mutation method	Random change a gene in an unstable student set	As indicated in Section 3.2
Number of Iterations	100	500

simulated annealing algorithms, how to design annealing strategies is an important issue. It might try different parameters to tune performance via a lot of experiments.

Yu and Sun (2016) have presented a fuzzy clustering method to solve a dormitory room assignment problem, which considered students' accommodation preferences. In this fuzzy clustering method, the dataset was the students with their accommodation preferences, and the number of cluster centers was equal to the number of rooms. Yu and Sun (2016) also conducted a simulation to validate their algorithm, where eight students were assigned to two rooms, and three accommodation preferences were considered: hygiene, bedtime, and sports habits.

Wang, Li, and Ma (2009) formulated the dormitory room assignment problem as a resource allocation problem, and utilized a backtracking algorithm to solve the problem. In this backtracking algorithm, students' preferences were considered as required conditions, and rooms as resources. The backtracking algorithm was a tree-structured architecture with nodes and branches. When preforming searching, a node in a searching path tried to find any possible child nodes for acceptable solutions. If it was impossible to fine any suitable child nodes for a particular node, the searching path went back to a previous node of this particular node. The searching space for this backtracking algorithm might be huge, when the numbers of students and rooms increase. This might result in long computational time to find the solutions.

Eslami and Afzali (2007) have presented a GA-based approach to solve a dormitory room assignment problem, which assigned rooms according to students' nationalities, religions, and course schedules. In the work by Eslami and Afzali, an experiment was conducted where 10 students were assigned to five rooms. The comparative table between the experiment in (Eslami and Afzali 2007) and our experiment is demonstrated in Table 1.

Experiment Design

Questionnaire

The participating students were 691 freshmen who entered a national university in southern Taiwan in the autumn semester of 2017. This study conducted an online survey for these freshmen. In the survey the freshmen were asked to answer five living habits and preferences with a one-to-one correspondence. Each question had three answers; the freshmen selected only one of the three answers for each question. In addition, the students were asked to give a degree of importance for each question, which was an integer between 1 and 10 (1 = least important; 10 = most important). The questions and their associated answers are demonstrated as follows:

(1) Do you prefer to live with the roommates from the same department as yours?

- Answer 1: I only want to live with the roommates from the same department as mine.
- Answer 2: I prefer to live with the roommates from the same department as mine. If it is impossible, I can accept living with the roommates from a different department.
- Answer 3: I don’t care whether or not my roommates are from the same department as mine.

(2) Do you prefer living with the roommates who will keep your room clean?

- Answer 1: I often clean my room, and hope that my roommates can do the same thing as I do.
- Answer 2: I occasionally clean my room, and hope that my roommates can do the same thing as I do.
- Answer 3: I don’t care whether or not my room is often cleaned. I only hope that my room can be kept in a basic cleanness situation.

(3) The interaction with my roommates

- Answer 1: I have an extroversion character, and I want my roommates can interact with me with a hustle and bustle way.
- Answer 2: I have an introversion character, and I want my roommates can interact with me with a quiet way.
- Answer 3: I don’t care whether my roommates interact with me in a hustle and bustle way, or in a quiet way.

(4) The preference of using an air-conditioning system (cooling system)

Answer 1: I often use an air-conditioning system, and hope that my roommates can do the same thing as I do.

Answer 2: I don't often use an air-conditioning system, and hope that my roommates can do the same thing as I do.

Answer 3: I don't care whether the air-conditioning is often used or not.

(5) My sleeping habit

Answer 1: I usually go to bed before 11 p.m.

Answer 2: I usually go to bed between 11 p.m. and 1 a.m.

Answer 3: I usually go to bed after 1 a.m.

(6) Preference of roommates' sleeping habit

Answer 1: I hope my roommates can sleep before 11 p.m.

Answer 2: I hope my roommates can sleep between 11 p.m. and 1 a.m.

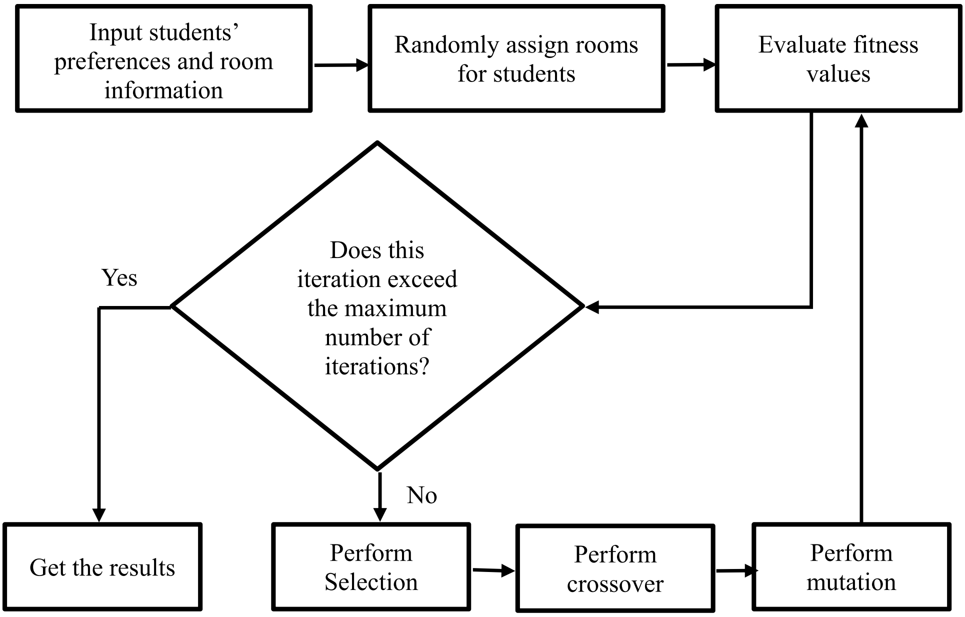

Input students' preferences and room information
Randomly assign rooms for students
Evaluate fitness values
Does this iteration exceed the maximum number of iterations?
Yes
No
Get the results
Perform Selection
Perform crossover
Perform mutation

Figure 1. The flowchart of the GA computing.

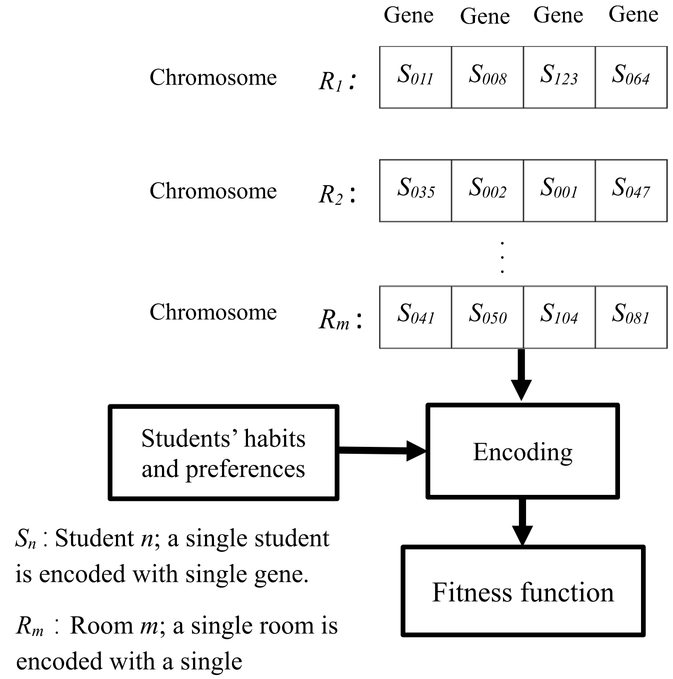

Gene
Gene
Gene
Gene
Chromosome R_1:
S_{011} S_{008} S_{123} S_{064}
Chromosome R_2:
S_{035} S_{002} S_{001} S_{047}
Chromosome R_m:
S_{041} S_{050} S_{104} S_{081}
Students' habits and preferences
Encoding
Fitness function
S_n : Student n; a single student is encoded with single gene.
R_m : Room m; a single room is encoded with a single

Figure 2. The encoding and structures of genes and chromosomes.

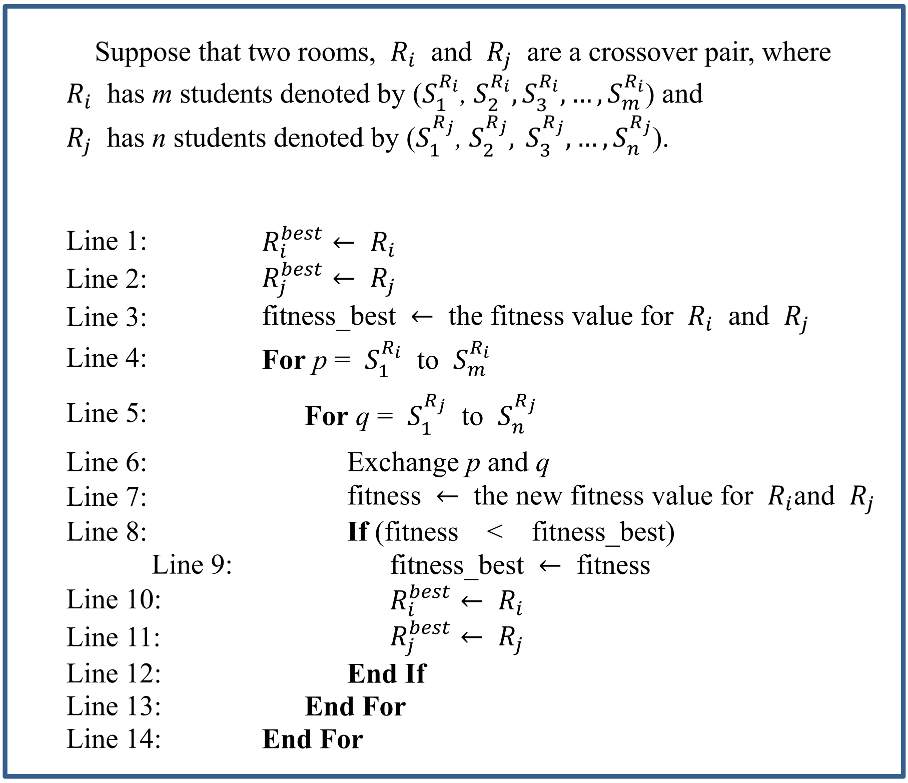
Answer 3: I hope my roommates can sleep lately, or I don't care the time my roommates go to bed.

GA Design

Figure 1 shows the flowchart of the GA computing used in this study. The encoding structures of genes and chromosomes are shown in Figure 2. In Figure 2, a single room is considered as a single chromosome. Each student in a room is considered as a gene encoded with his or her living habits and preferences. The encoded data are then fed into a fitness function to calculate fitness values. It is important to note that the size of a room (i.e., how many beds in a room) is equal to the student capacity of a room since a student just owns a bed in his or her room.

Crossover Method

In our real-world experiments, the rooms possibly had some unavailable beds. The sizes of chromosomes (i.e., beds of rooms) might be different. Therefore, we could not use traditional crossover methods (e.g., single-point crossover, two-point crossover, etc.) to perform the crossover operations during our experiments. We then used a special crossover



Suppose that two rooms, R_i and R_j are a crossover pair, where R_i has m students denoted by $(S_1^{R_i}, S_2^{R_i}, S_3^{R_i}, \ldots, S_m^{R_i})$ and R_j has n students denoted by $(S_1^{R_j}, S_2^{R_j}, S_3^{R_j}, \ldots, S_n^{R_j})$.

```
Line 1:   $R_i^{best} \leftarrow R_i$
Line 2:   $R_j^{best} \leftarrow R_j$
Line 3:   fitness_best $\leftarrow$ the fitness value for $R_i$ and $R_j$
Line 4:   **For** $p$ = $S_1^{R_i}$ to $S_m^{R_i}$
Line 5:       **For** $q$ = $S_1^{R_j}$ to $S_n^{R_j}$
Line 6:           Exchange $p$ and $q$
Line 7:           fitness $\leftarrow$ the new fitness value for $R_i$and $R_j$
Line 8:           **If** (fitness $<$ fitness_best)
Line 9:               fitness_best $\leftarrow$ fitness
Line 10:              $R_i^{best} \leftarrow R_i$
Line 11:              $R_j^{best} \leftarrow R_j$
Line 12:          **End If**
Line 13:      **End For**
Line 14:  **End For**
```

Figure 3. The pseudo code for the crossover operation.

method which randomly exchanged two students in two chosen rooms. The results of using this special crossover method led to unstable divergent results (i.e., the fitness values became worse and worse during the iteration process). We therefore used a locally exhaustive crossover method to solve this problem. In this method, for two parent chromosomes (rooms), we tried any possible combinations of genes for the two rooms, and found the best crossover result from all of the possible combinations of genes. Theoretically, the time complexity of the locally exhaustive crossover will be acceptable, if the sizes of two rooms are small. In this study, in most cases, the size of room was 4 (i.e., 4 beds in a room). The time consumption of using the locally exhaustive crossover method in our experiments was acceptable. We then claim that the locally exhaustive crossover method is suitable for this study.

To perform a crossover operation, two rooms from the parent chromosome pool are randomly selected as a crossover pair. We find exhaustively the best exchanging combination over all possible crossover pairs from the parent chromosome pool. The pseudo code for the crossover operation is shown in Figure 3. In this figure, lines 1 to 3 perform the initialization of parameters; lines 4 and 5 do a two-dimensional for

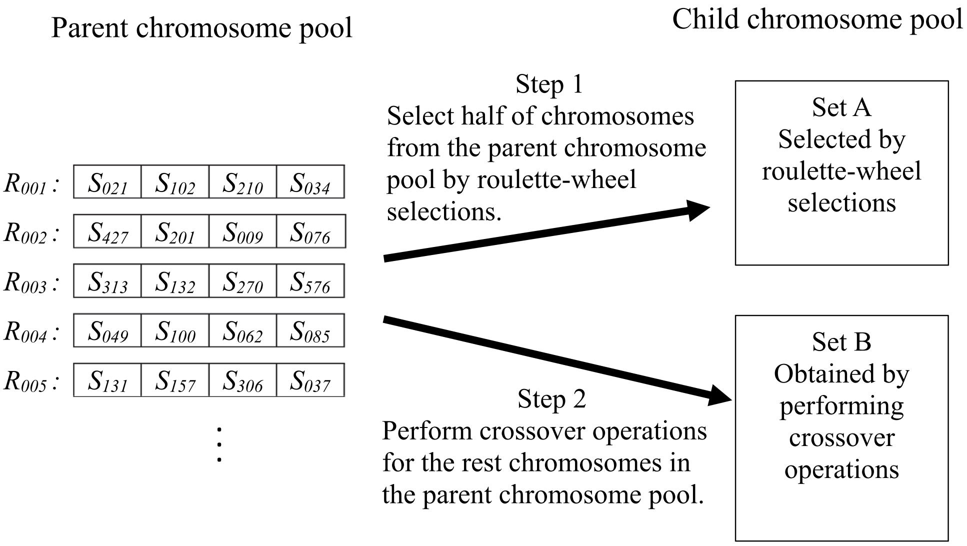

Parent chromosome pool
Child chromosome pool
R_{001}: S_{021} S_{102} S_{210} S_{034}
R_{002}: S_{427} S_{201} S_{009} S_{076}
R_{003}: S_{313} S_{132} S_{270} S_{576}
R_{004}: S_{049} S_{100} S_{062} S_{085}
R_{005}: S_{131} S_{157} S_{306} S_{037}
⋮
Step 1
Select half of chromosomes from the parent chromosome pool by roulette-wheel selections.
Step 2
Perform crossover operations for the rest chromosomes in the parent chromosome pool.
Set A
Selected by roulette-wheel selections
Set B
Obtained by performing crossover operations

Figure 4. The selection method.

looping operation; line 6 exchanges two students in R_i and R_j; line 7 recalculates the new fitness values for R_i and R_j after the exchanging done by line 6; lines 8 to 12 find the best fitness value, best R_i, and best R_j in the two-dimensional looping operation; lines 13 and 14 end the two-dimensional looping operation. After the crossover operation R_i and R_j are the best chromosomes among all possible crossover pairs.

Selection Method

At the beginning, we put all chromosomes (rooms) into the selection pool (i.e., the pool storing all parent chromosomes), and performed the locally exhaustive crossover method. The time consumption was huge. To reduce the computational time, we then used a half-half strategy when performing roulette-wheel selections. In this strategy, we first selected 50% of the chromosomes out of the entire selection pool using a roulette-wheel selection. These chosen chromosomes were reserved as seed chromosomes which would be automatically selected for the next iteration. The rest of chromosomes in the selection pool then performed the locally exhaustive crossover method to generate their offspring chromosomes for the next iteration. After using the half-half strategy, the computational time significantly decreased, and the performance (fitness values) was almost the same as the one which used the entire selection pool.

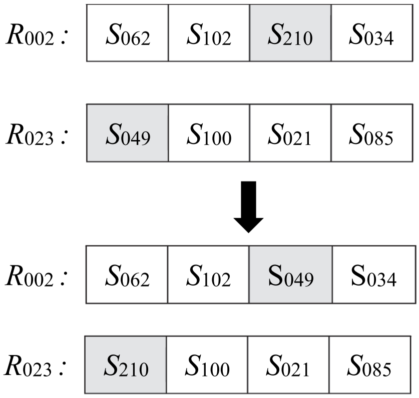

R_{002} : S_{062} S_{102} S_{210} S_{034}
R_{023} : S_{049} S_{100} S_{021} S_{085}
R_{002} : S_{062} S_{102} S_{049} S_{034}
R_{023} : S_{210} S_{100} S_{021} S_{085}

Figure 5. The mutation method.

The selection method is demonstrated in Figure 4. In this figure, the parent chromosome pool consists of all rooms. The child chromosome pool is established by two sets (Set A and Set B). The following steps are the procedure of performing the selection method:

Step 1: Forming Set A: use roulette-wheel selection operations to select half of the rooms from the parent chromosome pool.

Step 2: Forming Set B: perform crossover operations for the rest chromosomes in the parent chromosome pool.

Sets A and B form the child chromosome pool for the next generation.

Mutation Method

The following steps are the procedure of performing the mutation method:

Step 1: Randomly select two rooms (chromosomes).

Step 2: Randomly choose two students from the two chosen rooms, respectively.

Step 3: The two chosen students exchange each other.

Figure 5 shows an example of a mutation operation. In this figure, suppose that R_{002} and R_{023} are the two rooms randomly chosen for the mutation (Step 1). S_{210} and S_{049} are the two students randomly chosen from R_{002} and R_{023}, respectively (Step 2). During mutation, S_{210} and S_{049} exchange each other (Step 3). After mutation, S_{049} is in R_{002}, and S_{210} is in R_{023}. In Figure 5, the gray markers denote the changes before and after the mutation operation. In this study, the mutation rate is set to 0.002.

Termination Strategy

The maximum number of iterations is 500, i.e., the GA stops the procedure after 500 generations.

Fitness and Objective Functions

Consider a room set $R = \{s_1, s_2, \ldots, s_{N_s}\}$ where N_s is the number of students in this room. Let $Q = \{i_1, i_2, \ldots, i_5\}$ be a question set (the questionnaire) consisting of five items (questions) corresponding to the five living habits and preferences in the questionnaire, respectively.

The fitness function for R is formulated as follows:

$$f(R) = \left\{ \left[\sum_{\substack{x=1 \\ S_x \in R}}^{N_s} \left(\sum_{\substack{y=1 \\ S_y \in R}}^{N_s} \sum_{z=1}^{5} \left| P(S_x, A_z) - P(S_y, A_z) \right| \times D(S_x, I_z) \right) / N_s \right] / N_s \right\} + 1 \tag{1}$$

where

S_x :The x^{th} student in R.

S_y :The y^{th} student in R (S_y is considered the roommate of S_x).

$P(S_x, A_z)$:The answer of the z^{th} item in Q, done by S_x.

$P(S_y, A_z)$:The answer of the z^{th} item in Q, done by S_y.

$D(S_x, I_z)$:The degree of importance for the z^{th} question, which is specified by S_x. This degree is an integer ranging from 1 (least important) to 10 (most important).

$|\cdot|$: Absolute value operation.

N_s :The number of students in a room.

In Equation (1), $\left|P(S_x, A_z) - P(S_y, A_z)\right| \times D(S_x, I_z)$ is called a weighted preference distance from S_x to S_y since the preference distance $(\left|P(S_x, A_z) - P(S_y, A_z)\right|)$ between S_x and S_y is multiplied by importance degree $D(S_x, I_z)$. It is important to note that in Equation (1), 1 is added to this formula to avoid a divided-by-zero error. This error will occur, if all weighted distances are zeros. This will cause an infinite fitness value (un-computable value), when we perform a roulette-wheel selection.

The goal of this study is to minimize the overall fitness value for all rooms. Therefore, the objective function for the proposed GA is

$$\min \frac{\sum_{r=1}^{N_r} f(R_r)}{N_r} \tag{2}$$

where

R_r : The r^{th} room in the dormitory.

N_r : The number of rooms in the dormitory.

Experiments and Results

To validate the proposed GA, this study conducted three experiments. The computer settings for these experiments are shown as follows:

- Operation system:Windwos 10 (Virtual Machine)
- CPU:Interl Xeom CPU E5620 2.4 GHZ (4-core processor)
- RAM:16 GB DDR3-1333
- Hard disc drive:160 GB

Experiment 1: The Room Capacity Is Constant

In Experiment 1, each room had 4 beds (i.e., capacity of 4 students). Three cases were considered. The student sizes and the number of rooms for the three cases are shown as follows:

- Case 1: 33 students assigned to 9 rooms
- Case 2: 290 students assigned to 73 rooms
- Case 3: 691 students assigned to 173 rooms

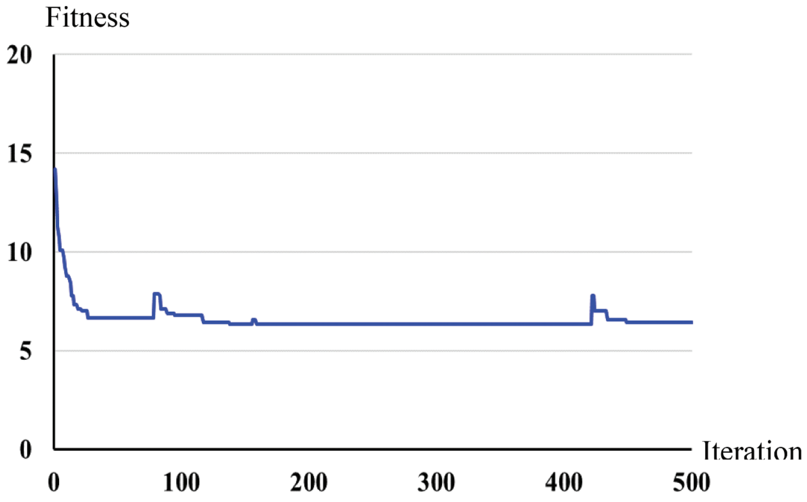


Figure 6. Iterative results of Case 1 (33 students assigned to 9 rooms).

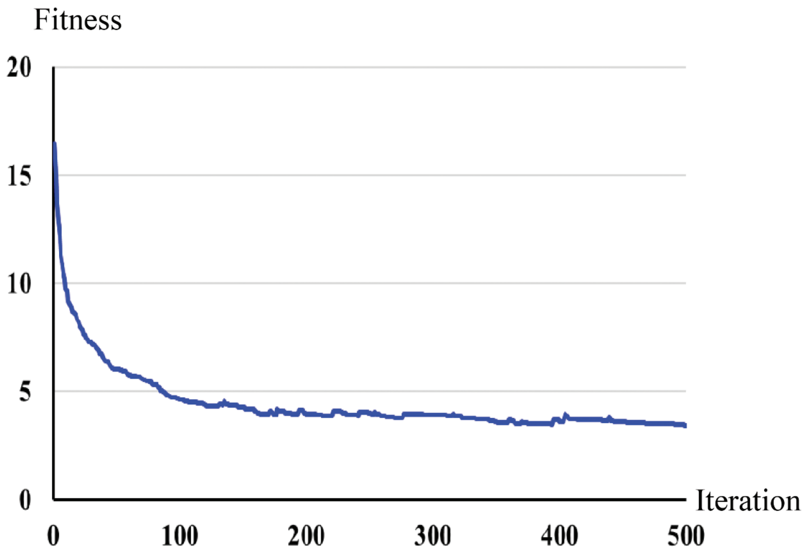


Figure 7. Iterative results of Case 2 (290 students assigned to 73 rooms).

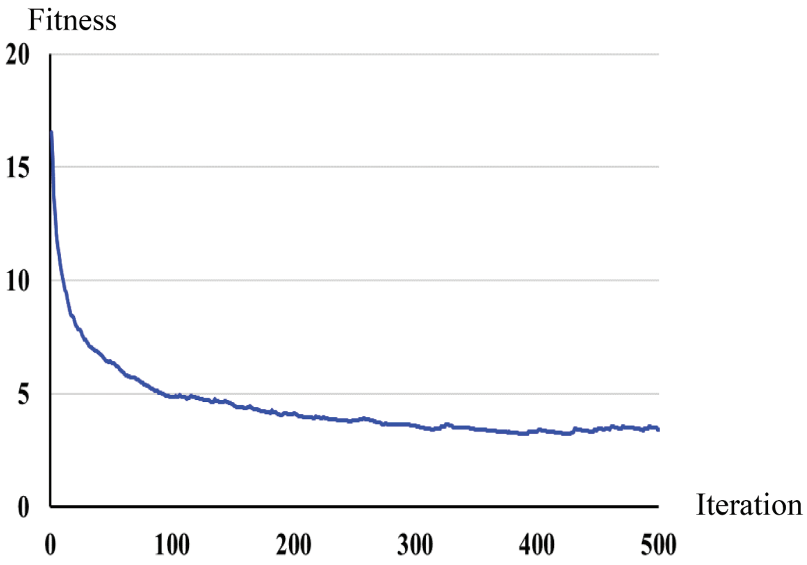


Figure 8. Iterative results of Case 3 (691 students assigned to 173 rooms).

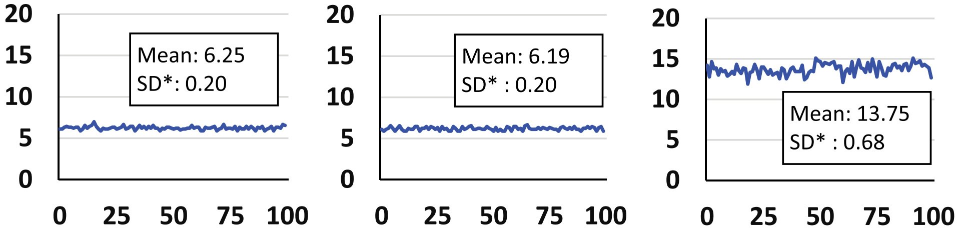
The iterative results of the three cases in Experiment 1 are shown in Figures 6, Figures 7, Figures 8. From these figures, we find that the more numbers the students and rooms have, the more smoothly the fitness decrease. In addition, two counterpart methods were used to evaluate the performance of the proposed algorithm: a simulated annealing method and a random assignment method. The comparative results are also demonstrated in Table 2. From this table, one concludes

Table 2. Comparative results of experiment 1.

	Case 1		Case 2		Case 3	
Method	Fitness Value	Execution Time*	Fitness Value	Execution Time*	Fitness Value	Execution Time*
The Proposed Algorithm	6.33	1	3.42	75	3.23	396
Simulated Annealing Method	6.22	9.13	4.49	582.54	3.40	3676.2
Random Assignment Method	14.22	N/A	16.47	N/A	16.57	N/A

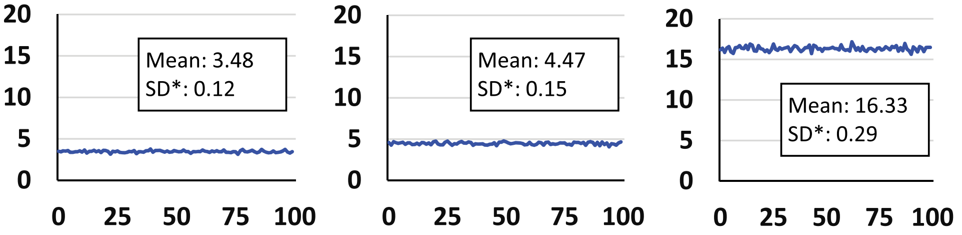
*: Calculated in second

- The execution time of the proposed algorithm is much faster than that of the simulated annealing method.
- The fitness value of the proposed algorithm is much better than that of the random assignment method.
- The fitness value of the proposed algorithm is almost the same as that of the simulated annealing method.



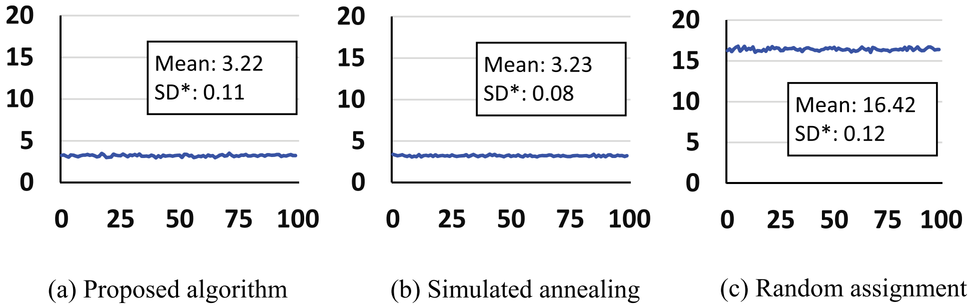
(a) Proposed algorithm (b) Simulated annealing (c) Random assignment

Figure 9. Stability test of Case 1: 33 students assigned to 9 rooms. Notes: x-axis: execution number; y-axis: fitness value; *: Standard Deviation.



(a) Proposed algorithm (b) Simulated annealing (c) Random assignment

Figure 10. Stability test of Case 2: 290 students assigned to 73 rooms. Notes: x-axis: execution number; y-axis: fitness value; *: Standard Deviation.



(a) Proposed algorithm (b) Simulated annealing (c) Random assignment

Figure 11. Stability test of Case 3: 691 students assigned to 173 rooms. Notes: x-axis: execution number; y-axis: fitness value; *: Standard Deviation.

Experiment 2: Stability of the Proposed Algorithm

To test the stability of the proposed algorithm, the procedure in Experiment 1 independently executed 100 times for each case. The results of the 100 executions for each case are shown in Figures 9, Figures 10, Figures 11, respectively. From these figures, for the proposed algorithm, the fitness values of the 100 executions for each case do not change sharply. The stability of the proposed algorithm is acceptable. For the simulated annealing method, the fitness values of the 100 executions for each case do not change sharply. For the random assignment method, the fitness values of the 100 executions for each case change a little bit sharply.

Experiment 3: Different Capacities of Rooms

In Experiment 3, the capacities (beds) of rooms were different. The room information for this experiment is shown as follows:

- Number of rooms consisting of 6 beds: 30
- Number of rooms consisting of 5 beds: 10
- Number of rooms consisting of 4 beds: 101
- Number of rooms consisting of 3 beds: 10
- Number of rooms consisting of 2 beds: 10
- Number of rooms consisting of 1 bed: 10

Figure 12 shows the iterative results of Experiment 3. From this figure, the fitness values stably decrease until approximately 350 generations. Again, the two counterpart methods were also used to evaluate the performance of the proposed algorithm. Table 3 shows the comparative

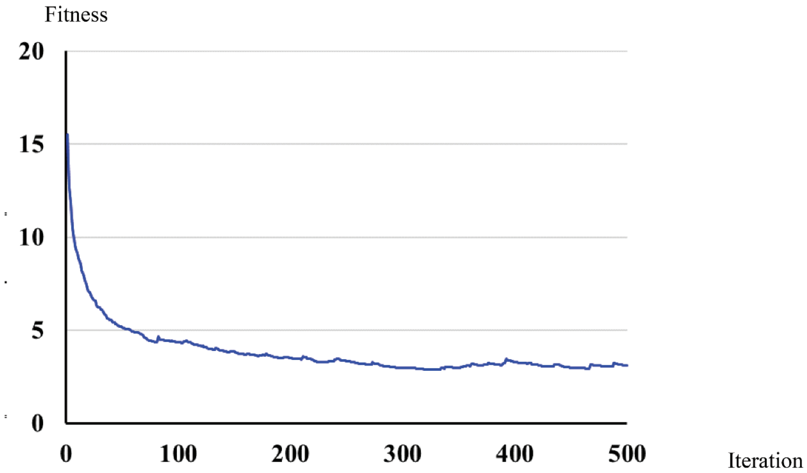


Figure 12. Iterative results of Experiment 3.

Table 3. Comparative results of experiment 3.

Method	Fitness Value	Execution Time*
The Proposed Algorithm	2.87	490
Simulated Annealing Method	2.87	3788
Random Assignment Method	15.51	N/A

*: Calculated in second

results of using the proposed algorithm, the simulated annealing method, and the random assignment method. These results are similar to those of Experiment 1, i.e.,

- The execution time of the proposed algorithm is significantly less than that of the simulated annealing method.
- The fitness value of the proposed algorithm is significantly less than that of the random assignment method.
- The fitness value of the proposed algorithm is almost the same as that of the simulated annealing method.

Conclusions

Dormitory room assignment is essentially a stable roommates problem. Some previous studies discussed this problem from the viewpoint of mathematics. The searching space for solving the dormitory room assignment is huge when the numbers of students and rooms become large. To reduce the searching space, this study proposed a GA based algorithm to find sub-optimal solutions with reasonable computational time. This study considered five living habits and preferences when assigning dormitory rooms.

Unlike popular crossover methods and selection methods used in traditional GAs, this study has presented a modified crossover method (i.e., locally exhaustive crossover method) and a modified selection method (i.e., half-half selection strategy). In the locally exhaustive crossover method, for two chromosomes (rooms), we tried any possible gene's combinations for the two rooms, and found the best crossover result. This could avoid divergence results during the GA's iterations, and thus got better fitness values for this dormitory assignment problem. In the half-half selection strategy, instead of using the entire selection pool, we selected the half of chromosomes from the entire selection pool using a roulette-wheel selection method; the other half of chromosomes then performed the locally exhaustive crossover method to get their offspring chromosomes. This would significantly reduce the time consumption during the GA's iterations, and the performance (fitness values) was almost the same as the one that used the entire selection pool.

This study also conducted experiments to validate the proposed algorithm. Experimental results have shown that the proposed algorithm could have acceptable performances with reasonable computation time. In addition, two counterpart methods were used to evaluate the performance of the proposed algorithm: a simulated annealing method and a random assignment method. The comparative results have suggested that

- The execution time of the proposed algorithm is much faster than that of the simulated annealing method.
- The fitness value of the proposed algorithm is much better than that of the random assignment method.
- The fitness value of the proposed algorithm is almost the same as that of the simulated annealing method.
- The proposed algorithm is stable to repeated executions.
- The proposed algorithm is still suitable even when the capacities (beds) of rooms are different.

The future studies might possibly address the following directions:

- Adding more living habits and preferences.
- Increasing the numbers of students and rooms.
- Optimizing the computational time.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

ORCID

Che-Chern Lin http://orcid.org/0000-0003-1142-5891

References

Abraham, D. J., P. Biró, and D. F. Manlove. 2005. Almost stable' matchings in the roommates problem. *Approximation and Online Algorithms Lecture Notes in Computer Science* 3879:1–14.

Albalawi, F. O., and M. S. Maashi. 2021. Selection and optimization of software development life cycles using a genetic algorithm. *Intelligent Automation & Soft Computing* 28 (1):39–52. doi:10.32604/iasc.2021.015657.

Alimudin, A., I. Winarno, and Y. Ishida. 2019. Defining network switch preference by network traffic using stable marriage problem. Paper presented at the 2019 International Electronics Symposium (IES), Surabaya, Indonesia, September 27-28.

Biró, P., E. Iñarra, and E. Molis. 2016. A new solution concept for the roommate problem: Q-stable matchings. *Mathematical Social Sciences* 79:74–82. doi:10.1016/j.mathsocsci.2015.12.001.

Bristi, W. R., F. Chowdhury, and S. Sharmin. 2019. Stable matching between house owner and tenant for developing countries. Paper presented at the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, July 6-8.

Chang, -C.-C. (2019). Applying a genetic algorithm to dormitory assignment optimization. Master Thesis, National Kaohsiung Normal University, Kaohsiung, Taiwan.

Chen, R., M. Chen, and H. Yang. 2020. Dynamic physician-patient matching in the healthcare system. Paper presented at the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, July 20-24.

Eslami, H., and M. Afzali. 2007. Applying genetic algorithm for allocating students to the dormitory rooms. Iran Data Mining Conference, Tehran, Iran, November 20, 2007, 1–8.

Gale, D., and L. S. Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69 (1):9–15. doi:10.1080/00029890.1962.11989827.

Irving, R. W. 1985. An efficient algorithm for the 'stable roommates' problem. *Journal of Algorithms* 6 (4):577–95. doi:10.1016/0196-6774(85)90033-1.

Irving, R. W., and D. F. Manlove. 2002. The stable roommates problem with ties. *Journal of Algorithms* 43 (1):85–105. doi:10.1006/jagm.2002.1219.

Iwama, K., and S. Miyazaki. 2008. A survey of the stable marriage problem and its variants. Paper presented at the International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008), January 17-17.

Katarína, C., and S. Ferková. 2004. The stable crews problem. *Discrete Applied Mathematics* 140 (1–3):1–17. doi:10.1016/j.dam.2003.05.003.

Le Hong, T., V. Hoang Huu, and T. Chung. 2016. Finding "optimal" stable marriages with ties via local search. Paper presented at the 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE), Hanoi, Vietnam, October 6-8.

Madi, K., E. Paquet, and H. Kheddouci (2019). New graph distance based on stable marriage formulation for deformable 3D objects recognition. Paper presented at the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, UAE, November 3–7.

Meulbroek, D., D. Ferguson, M. Ohland, and F. Berry. 2019. Forming more effective teams using CATME teammaker and the gale-shapley algorithm. Paper presented at the 2019 IEEE Frontiers in Education Conference (FIE), October 16-19.

Nakamura, M., K. Onaga, S. Kyan, and M. Silva. 1995. Genetic algorithm for sex-fair stable marriage problem. Paper presented at the Proceedings of ISCAS'95 - International Symposium on Circuits and Systems, April-May 30-3.

Pittel, B. G., and R. W. Irving. 1994. An upper bound for the solvability probability of a random stable roommates instance. *Random Structures and Algorithms* 5 (3):465–86. doi:10.1002/rsa.3240050307.

Ren, J., F. Xia, X. Chen, J. Liu, M. Hou, A. Shehzad, N. Sultanova, X. Kong, et al. 2021. Matching algorithms: fundamentals, applications and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*. 5(3):332–50. doi:10.1109/TETCI.2021.3067655.

Roth, A. E., T. Sönmez, and M. U. Ünver. 2005. Pairwise kidney exchange. *Journal of Economic Theory* 125 (2):151–88. doi:10.1016/j.jet.2005.04.004.

Settanni, E. (2000). Improving dorm room assignments using simulated annealing, *Master Thesis*, The University of New Mexico.

Tan, J. J. M. 1990. A maximum stable matching for the roommates problem. *Bit* 30 (4):631–40. doi:10.1007/BF01933211.

Trang, L. H., N. T. Hoa, T. V. Hoai, and H. H. Viet. 2019. Finding MAX-SMTI for stable marriage with ties and bounded preference lists. Paper presented at the 2019 International Conference on Advanced Computing and Applications (ACOMP), November 26-28.

Trung, N. T., and D. T. Anh. 2009. Comparing three improved variants of simulated annealing for optimizing dorm room assignments. 2009 IEEE-RIVF International Conference on Computing and Communication Technologies, Da Nang, Vietnam, 1–5.

Vien, N. A., and T. Chung. 2006. Multiobjective fitness functions for stable marriage problem using genetic algrithm. Paper presented at the 2006 SICE-ICASE International Joint Conference, Busan, Korea, October 18-21.

Wang, W.-F., Z.-L. Li, and Y. Ma. 2009. Application of backtracking algorithm in college dormitory assignment management. 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, August 8-11, 2009, 272–74.

Xiao, Y., D. Niyato, K. Chen, and Z. Han. 2016. Enhance device-to-device communication with social awareness: A belief-based stable marriage game framework. *IEEE Wireless Communications* 23 (4):36–44. doi:10.1109/MWC.2016.7553024.

Yu, Q., and X.-A. Sun. 2016. A research on the model of university apartment occupancy distribution based on the student preferences. International Conference on Logistics, Informatics and Service Sciences (LISS), Sydney, Australia, 24-27 July, 2016, 1–5.

Zayed, H. A. B., and M. S. Maashi. 2021. Optimizing the software testing problem using search-based software engineering techniques. *Intelligent Automation & Soft Computing* 29 (1):307–18. doi:10.32604/iasc.2021.017239.