Automated Scheduling of Hostel Room Allocation Using Genetic Algorithm

Rayner Alfred and Hin Fuk Yu

Abstract Due to the rapid growth of the student population in tertiary institutions in many developing countries, hostel space has become one of the most important resources in university. Therefore, the decision of student selection and hostel room allocation is indeed a critical issue for university administration. This paper proposes a hierarchical heuristics approach to cope with hostel room allocation problem. The proposed approach involves selecting eligible students using rank based selection method and allocating selected students to the most suitable hostel room possible via the implementation of a genetic algorithm (GA). We also have examined the effects of using different weight associated with constraints on the performance of the GA. Results obtained from the experiments illustrate the feasibility of the suggested approach in solving the hostel room allocation problem.

Keywords Scheduling · Hostel space · Genetic algorithm

1 Introduction

A hostel management system is responsible for allocating students to available hostel spaces and managing students' resident and hostels information. The rapid rise in the student population of the tertiary institutions over the years in developing countries has become an inevitable challenge to university administration due to limited hostel spaces available in the university. Therefore, in order to cope with the issues above, optimization of the scheduling process for hostel room allocation is a must. Hostel room scheduling is the problem of allocating eligible students to hostel rooms while satisfying specified constraints and the scheduling process is

R. Alfred (✉) · H. F. Yu
Knowledge Technology Research Unit, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia
e-mail: ralfred@ums.edu.my

H. F. Yu
e-mail: roronuozoro@gmail.com

N. Sharma et al. (eds.), *Data Management, Analytics and Innovation*, Advances in Intelligent Systems and Computing 1042, https://doi.org/10.1007/978-981-32-9949-8_11

automated [2]. Automated scheduling is usually executed using a scheduling algorithm to generate a sequence of actions in order to achieve certain objectives. Some of the applications or problems which require automated scheduling are flow shop scheduling problem (FSSP) and job shop scheduling problem (JSSP) [1] including Hostel Space Allocation Problem [2]. Some examples of technique or method for dealing with scheduling problem are hybrid metaheuristic algorithm [3], relax and fix heuristics [4], and hierarchical heuristics approach [2]. The reason why these proposed techniques are advantageous is mainly because of the simplicity, flexibility, inexpensive computational cost, and derivation-free mechanism of heuristic or metaheuristic [5]. In this paper, a hierarchical heuristics approach will be proposed and implemented to optimize the scheduling process of hostel room allocation. Hierarchical heuristics approach is chosen because it is able to provide good results in dealing with multilevel allocation process of hostel space allocation problem [2].

The rest of the paper is organized as follow. Section 2 discusses related works with regard to implementation of GA-based heuristic in dealing with scheduling problems. Section 3 presents the design of the proposed hierarchical heuristics approach while the experimental setup is discussed in Sect. 4. This paper is concluded in Sect. 5.

2 Related Works

A large amount of research and study has been conducted in a variety of domains of scheduling problem such as JSSP [6], shelf space allocation problem (SSAP) [7], timetabling problem [8], knapsack problem [9], and bin-packing problem [10]. Yaqin et al. [6] studied on the job shop scheduling with multiobjectives-based on GA and according to the research, GA is relatively faster than other methods used for scheduling which are usually based on single point search as GA do multipoint searches on the whole population simultaneously. Their studies show that GA is feasible in optimizing NP-hard or combinatorial optimization problem. Castelli and Vanneschi [7] proposed a hybrid algorithm that combines GA and variable neighbor search (VNS) algorithm to increase the explorative ability of GA to prevent GA from being stuck in local optimum and results obtained from their experiments had shown the enhancement in explorative ability of GA. Yang et al. [9] introduced attribute reduction of rough sets into the crossover of GA to tackle multidimensional knapsack problem which is also known as the multi-constraints knapsack problem. The steps of the GA proposed by them are different than the steps in typical GA. If reduct is found in the genes of the parent chromosome, the particular genes in the reduct will be the points selected to crossover whereas in the event of no reduct is found, the single-point crossover is applied. Results obtained from their studies show that the likelihood of the proposed GA to reach a maximal solution is relatively higher than conventional GA. Other than that, Bennell et al. [10] proposed a multi-crossover genetic algorithm (MXGA) to solve a new variant of the two-dimensional bin-packing problem where each rectangle is assigned a due date and each bin has a fixed processing time. Experiments conducted in their research shows that the MXGA performance better than the single crossover genetic algorithm in solving the bin-packing problem. One crucial similarity between the works reviewed above is such that GA is always a part of the proposed solution or method. Therefore, the motivation of our work is to implement GA in the proposed hierarchical heuristic approach in order to tackle the hostel room allocation problem. The hierarchical heuristics approach proposed in this work is different than the hierarchical heuristics approach proposed by Adewumi and Ali [2], in which instead of greedy-like heuristic, the rank based selection method is used in the selection stage. Moreover, the representation, as well as crossover and mutation mechanism of the GA, is different as well.

3 Hierarchical Heuristics Approach

This section explains the framework of the hierarchical heuristics approach. The main components of the approach, which are rank-based selection method and GA will be explained. The proposed hierarchical heuristics approach framework is shown in Fig. 1.

With respect to Fig. 1, necessary information of applicants, rooms and specified constraints will initially be provided to the framework. This information will be used during the selection stage; with the implementation of rank-based selection method, applicants will be ranked based on the weight they hold and higher ranker will be selected first. Weight hold by the applicant is determined via the degree in which the applicant satisfies the specified constraints and the order of priority of constraints will be shown in Sect. 4. A list of selected students or applicants will be produced at the end of the selection stage. During the allocation stage, the selected students will be allocated to the most suitable hostel room possible using GA-based on specified constraints. The constraints will be shown in Sect. 4.

3.1 Genetic Algorithm

A GA is a stochastic search algorithm which can be used to a variety of combinatorial optimization problems [11]. GA is also known as nature inspired algorithm for the reason that the steps in GA are based on the evolutionary process of biological organisms. GA evolves according to the principle “survival of the fittest”; this implies that offspring that carries best attributes genes will have higher chance to survive to the next generation. Generally, GA simulates the evolutionary process of biological organisms by generating individuals randomly that form up an initial population. Each individual is encoded into a string (chromosome) which represents a possible solution to a given problem. Fitness function is performed on

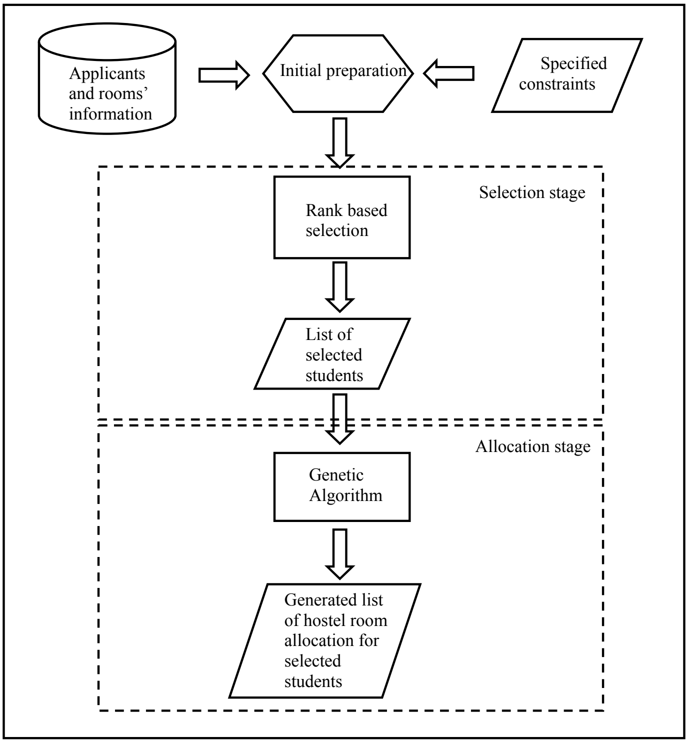


Fig. 1 Framework of hierarchical heuristics approach

each and every individual and each individual has its own fitness value. A probability of being selected during the selection process is assigned to each individual based on the individual's fitness value so as to ensure the characteristics of highly fit individuals or chromosomes are passed down to the next generation through a crossover procedure. The most general crossover method is called a one-point crossover; in which a single crossover point on both parents' chromosomes is selected first and then all information located beyond that single crossover point in either chromosome are swapped between these two parent chromosomes. The mutation is often applied to the offspring's chromosomes after crossover by altering some genes in the chromosomes, but with very low occurrence rate. In most of the case, less fit individuals will be replaced by the highly fit offspring and the new population will be reevaluated by the fitness function again. This sequence of processes will be repeated until a termination criterion is met.

3.2 Solution Encoding and Generic Operators

The encoding applied to represent candidate solutions for the GA used in the allocation stage as shown in Fig. 1 will be described in this section. The generic operators, which are crossover and mutation of the GA, will be described as well. Figure 2 illustrates how the candidate solutions are encoded. Based on Fig. 2, the integer values are the unique ID of selected students. The number of genes owned by each room varies with the types of room and as for this case, single room owns two genes while double room owns four genes. The length of the chromosome varies with the total number of rooms available. Figure 3 illustrates the process of crossover.

In Fig. 3, a self-crossover is used in order to ensure there is no repetition of integer values in the chromosome as the integer values are the unique ID of students. The crossover is done by randomly selects two portions with the same number of genes and swaps the genes of the selected portions in a random manner. The genes swapping phenomenon is controlled by a crossover rate of 0.45. On the other hand, the mutation process is almost the same as crossover process, in which the difference is that all the genes of the selected portions will be swapped completely in a random manner instead of controlled by a crossover rate of 0.45. The likelihood of the mutation process to happen is controlled by a mutation rate of 0.05.

3.3 Fitness Function

Fitness function, which is also known as evaluation function is used to evaluate individuals after initialization, crossover, and mutation steps to determine how fit the individuals are. In this work, the fitness of GA's individual will be evaluated based on the degree in which the constraints are satisfied. Each individual of the GA will have their own fitness value which is a set of real number in [0, 1]. A value of 0 indicates a complete violation of all given constraints while a value of 1 indicates no constraint is violated. The fitness of an individual, by referencing the fitness function in [2], is computed as:

$$f = \sum_{i=1}^{n} w_i u_i \in [0,\ 1] \tag{1}$$

where u_i in Eq. (1) is the utilization factor given as:

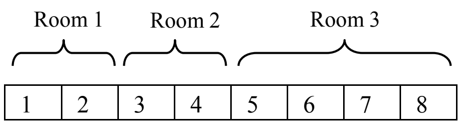
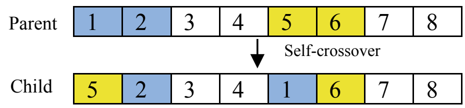


Fig. 2 Chromosome encoding of the GA in allocation stage

Fig. 3 Crossover process



$$u_i = \frac{1}{n} \tag{2}$$

w_i is the weight hold by gene i in a chromosome, n in Eqs. (1) and (2) is the total number of genes a chromosome has.

4 Experimental Setup

This section describes the experiments that are carried out in this work. The sample datasets for students (applicants) and rooms are shown in Tables 1, 2 and 3. The initial population of the GA will be randomly generated.

The main parameters for GA used are as follow 30 population size, 100 generations, crossover rate at 45% and mutation rate at 5%. The constraints applied to the selection and allocation process are as follow:

1. Selection:
   - Hard constraints
     - All first year students must be selected (FY).
     - Student representative must be selected (SR).
     - Uniformed units must be selected (SU).
     - Handicapped students must be selected (HS).
   - Soft constraints
     - Students with higher student development points are more likely to be selected (SD).
     - Students with higher CGPA are more likely to be selected (SC).

Table 1 Statistics of applicants per category

Category	Male	Female
First year	703	2635
Senior	297	1115
Student representative	4	8
Uniformed units	197	353
Handicapped	2	5
Food science and forestry	192	328
Others	605	3056

Table 2 Availability of rooms for male students

	Hostel				
Type	A/B	C/D	E	Angkasa	Usia
Single room capacity	52	52	52	52	
Double room capacity	148	148	148	148	0
Ground floor capacity	50	50	50	50	0

Table 3 Availability of rooms for female students

	Hostel				
Type	A/B	C/D	E	Angkasa	Usia
Single room capacity	140	140	140	140	140
Double room capacity	560	560	560	560	560
Ground floor capacity	150	150	150	150	150

2. Allocation:

- Hard constraints
  - Uniformed units must be allocated to hostel E (UH).
  - Handicapped students must be allocated to rooms situated on the ground floor (HG).
  - Single rooms are for senior students (SS).
- Soft constraints
  - Food Science (FS) and Forestry (FT) student should be allocated to hostel E (FE).
  - Students should be allocated to the hostel of their choices (SH).
  - First year students should be allocated to hostels inside the campus (SI).

Weights associated with selection and allocation constraints are shown in Tables 4 and 5. With respect to Table 4, weight associated to constraint FY is 0.51 and this is to make sure the sum of weights of other constraints is less than 0.51 as FY holds the highest priority in hard constraints. As for SR, SU, and HS, they have the same level of priority, in which fulfilling one or two or all of these constraints will give the student the same amount of weight, which is 0.25. This is to ensure the sum of weights for remaining soft selection constraints will not exceed 0.25. On the other hand, the four sets of weights' distribution for allocation constraints shown in Table 5 are used to examine the effect of weight associated constraints on the performance of the GA; during allocation stage, the results obtained from the rank based selection are processed by the GA using these four sets of weights' distribution and the results obtained are used as a basis for optimizing the weights' distribution for allocation constraints. The optimization of weights' distribution for allocation constraints is done by simulating the GA 30 times using the same set of results obtained from rank-based selection but with 30 different sets of weights' distribution for allocation constraints which are randomly generated.

The results obtained from the rank-based selection during selection stage are shown in Table 6 where it shows the fulfillment of hard constraints as soft constraints of selection, which are SD and SC cannot be quantified. However, with the implementation of rank-based selection and normalization of student development point and CGPA, SD, and SC will always be optimized. The optimized weights' distribution for allocation constraints is shown in Table 7 while the allocation results obtained from simulating the GA using the four sets of weights' distribution and optimized weights' distribution are shown in Table 7. Based on Table 8, the performance of the GA is measured in term of constraints satisfaction in percentage

Table 4 Weights' distribution for selection constraints

Type	Constraints	Weight
Hard	FY	0.51
	SR, SU or HS	0.25
Soft	SD	0.12
	SC	0.12

Table 5 Weights' distribution for allocation constraints

Type	Constraints	Set 1	Set 2	Set 3	Set 4
Hard	UH	0.22	0.25	0.27	0.30
	HG	0.22	0.25	0.27	0.30
	SS	0.22	0.25	0.27	0.30
Soft	FE	0.12	0.09	0.07	0.04
	SH	0.11	0.08	0.06	0.03
	SI	0.11	0.08	0.06	0.03

Table 6 Selection results obtained from rank-based selection method

Category	Selected		Not selected	
	Male	Female	Male	Female
First year	703	2635	0	0
Senior	97	865	200	250
Student representative	4	8	0	0
Uniformed units	197	353	0	0
Handicapped	2	5	0	0
Food science and forestry	103	207	89	121
Others	494	2927	111	129

Table 7 Optimized weights' distribution for allocation constraints

Type	Constraints	Weight
Hard	UH	0.231
	HG	0.243
	SS	0.216
Soft	FE	0.098
	SH	0.109
	SI	0.105

Table 8 Allocation results obtained Using GA with different weights' distribution

Type	Constraints	No of students	Results				
			Set 1	Set 2	Set 3	Set 4	Optimized weight
Hard	UH	550	462	505	509	512	506
	HG	7	4	5	5	6	6
	SS	962	714	801	832	825	822
Total		1519	1180 (77.68%)	1311 (86.31%)	1346 (88.61%)	1343 (88.41%)	1334 (87.82%)
Soft	FE	310	255	202	181	172	251
	SH	3338	2890	2458	2231	2091	2718
	SI	4300	3457	3321	2989	2786	3362
Total		7948	6602 (83.06%)	5981 (75.25%)	5401 (67.95%)	5049 (63.52%)	6331 (79.65%)

instead of average fitness value of GA's population. This is because fitness value is based on weights' distribution for allocation constraints and, therefore, it is not suitable for indicating the performance of the GA. The results obtained from both selection and allocation stages show the feasibility of the suggested approach in solving the hostel room allocation problem.

5 Conclusion

This paper proposed a hierarchical heuristics approach that includes the implementation of GA to cope with hostel room allocation problem. The effect of weights associated with constraints is examined by simulating GA with multiple sets of weights' distribution for allocation constraints. Optimized weights' distribution for allocation constraints is obtained during the simulation process. Results obtained from the experiments conducted show the feasibility of the proposed approach in dealing with the problem at hand. However, improvements can still be made in the future to increase the overall performance of the proposed approach by tuning the GA components and parameters such as encoding method, crossover method, mutation method, crossover rate, mutation rate, population size, etc.

References

1. Zhang, X-F., Koshimura, M., Fujita, H., Hasegawa, R.: Combining PSO and local search to solve scheduling problems, pp. 347–354 (2011)
2. Adewumi, A.O., Ali, M.M.: A multi-level genetic algorithm for a multi-stage space allocation problem. Math. Comput. Model. **51**(1), 109–126 (2010)
3. Zhang, Q., Manier, H., Marie-Ange, M.: A hybrid metaheuristic algorithm for flexible job-shop scheduling problems with transportation constraints, pp. 441–448 (2012)
4. Deisemara, F., Reinaldo, M., Socorro, R.: Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants. Comput. Oper. Res. **37**, 684–691 (2009)
5. Seyedali, M., Andrew, L., Sanaz, M.: Confidence measure: a novel metric for robust meta-heuristic optimization algorithms. Inf. Sci. **317**, 114–142 (2015)
6. Yaqin, Z., Beizhi, L., Lv, W.: Study on job-shop scheduling with multi-objectives based on genetic algorithms, vol. 10, pp. 10–294 (2010)
7. Castelli, M., Vanneschi, L.: Genetic algorithm with variable neighborhood search for the optimal allocation of goods in shop shelves. Oper. Res. Lett. **42**(5), 355–360 (2014)
8. Soria-Alcaraz, J., Carpio, M., Puga, H.: A new approach of design for the academic timetabling problem through genetic algorithms, pp. 96–101 (2010)
9. Yang, H., Wang, M., Chen, Y., Huang, Y., Kao, C.: Crossover based on rough sets-a case of multidimensional knapsack problem, pp. 2411–2415 (2010)
10. Bennell, J., Soon Lee, L., Potts, C.: A genetic algorithm for two-dimensional bin packing with due dates. Int. J. Prod. Econ. **145**(2), 547–560 (2013)
11. Reeves, C.R.: Modern heuristic techniques for combinatorial problems. Blackwell Scientific, Hoboken (1993)