A Multi-level Genetic Algorithm for a Multi-stage Space Allocation Problem

Aderemi O. Adewumi[1], M. Montaz Ali[1], Joshua O. Ayeni[2]

[1]*School of Computational and Applied Mathematics, University of Witwatersrand, Johannesburg, South Africa*

[2]*Department of Computer Sciences, Faculty of Science, University of Lagos, Lagos, Nigeria*

Telephone: +27 11 717 6102

Fax: +27 11 717 6149

Email: laremtj@gmail.com

Abstract. We considered the case of a multi-stage hostel space allocation problem based on data set obtained from a tertiary institution. Genetic Algorithm was applied at different levels of the allocation. We studied the effects of parameter change on solutions obtained. The rate of feasibility of the solutions was also determined.

Keywords: *Space Allocation, Hostel Space Allocation, Combinatorial Optimization, Genetic Algorithm*

Abbreviations: SAP: Space Allocation Problem; GA: Genetic Algorithm; COP: Combinatorial Optimization Problem; HSAP: Hostel Space Allocation Problem

1 Introduction

Space allocation problem (SAP) is a domain specific combinatorial optimisation problem (COP) that has attracted attention among researchers recently (Bai 2005; Landa 2003). The problem is similar to scheduling problem such as academic timetabling (Adewumi et. al. 2003 & 2005) with its associated resource efficiency issues. SAP, in a higher institution context, is defined as the allocation of entities (staff, students, laboratories, lecture rooms, etc.) to available rooms in order to satisfy both hard and soft constraints (Landa 2003). It finds relevance in various fields such as storage space allocation on disk and office space allocation. More importantly, the distribution of the available space among staff, students and other demanding entities is a dynamic and continual exercise. In most real life cases with large instances, the use of proven optimization technique becomes inevitable. Hostel space allocation problem (HSAP) arises from the need to allocate available bed spaces distributed around various halls of residence (or hostels) among large number of demanding and eligible categories of students based on a set of given (and sometimes conflicting) constraints. The number of applicants and constraints for each category varies. The objective is to ensure optimal space utilization that satisfies given constraints. The complexity of the problem emanates from the limited number of bed spaces available, increasing demand from students and the conflicting nature of some constraints imposed on the allocation process. The best practice in most Nigerian universities is to use database or spreadsheet applications without recourse to any algorithm that determines an optimal allocation. Burke and Varley (1998a and b) suggested the application of heuristics to SAP.

Exact methods have been applied to various (and in most cases smaller) instances of real-life SAP. These include the use of mixed-integer goal programming (Ritzman et al. 1980), linear programming (Benjamin et. al. 1992), and integer goal programming (Giannikos et al., 1995). Like other NP-Hard COPs, exact algorithms for SAP have exponential time complexity (Landa 2003). For large instances of SAP and faster results, heuristics and their variants have been applied lately. Bai (2005) investigated the application of different metaheuristics to real-world shelf SAP that arise due to conflict of limited shelf space availability and the large number of products to be displayed. Landa (2003) investigated the application of metaheuristics to real-life instances of office SAP. These studies reveal, among other things, 1) the multi-objective nature of SAP, 2) the complex nature of SAP as a result of conflicting objectives and constraints, and 3) the critical nature of the real-world problems under space allocation. Experimental results obtained show the efficiency of heuristics in handling SAPs.

In this paper, we employ a genetic algorithm (GA) (Goldberg 1989) metaheuristic to solve the HSAP under domain specific constraints. This work is based on the multi-stage real-life HSAP with data set obtained from one of the largest university in Nigeria. To the best of our knowledge, this is the first application of heuristic to the problem.

2 Problem Definition (HSAP)

The allocation of bed spaces to various categories of students is performed by the Students Affairs Units at the beginning of each academic session. Due to inadequacy of space, only about one-fifth of the students' population of over 35,000 can be conveniently accommodated in the campus hostels. The University has major twelve halls of residence built in the main campus to accommodate the undergraduate students. These consist of six male hostels and six female hostels. The hostels are zoned based on their physical location (Tables 1). The hostels have varying number of rooms spread across varying number of floors on varying number of blocks. The rooms have varying capacity in terms of the number of bed spaces available therein.

The allocation process is multi-stage. First, various categories of students are distributed to available hostels by the Student Affairs Office based on given set of constraints. Next, the distribution list is used by each Hall Manager to allocate individual student under each category into a particular room in a block and floor based on another set of constraints. There are eight categories of students with differently assigned priority. These are 1) final year students, 2) Scholars (those with cumulative grade point average (CGPA) ≥ 4.20), 3) foreign students 4) physically challenged students or those with peculiar health conditions, 5) first year students, 6) sports men and women, 7) discretionary and 8) Others. The allocation processes for male and female students are mutually exclusive.

As in other COPs, HSAP is highly constrained. The first stage of allocation involves distributing categories of students into the various hostels so that the following hard constraints are satisfied: 1) allocation of certain categories to specified halls is satisfied (Table 2); 2) all eligible foreign students are allocated; 3) all students with record of peculiar health conditions are allocated; and 4) all sports men and women are allocated. For each hostel, the second stage distributes the resultant allocation list of the stage one into various rooms in the hostel such that the following hard constraints are satisfied: 1) all health students are allocated the lowest possible floor, and 2) allocated final year students are placed on the highest possible floor.

Table 1 – Capacities of halls of residence (Source DSA Office)

Zone (Area)	Name of Hostel	Sex	Hall Capacity
A (Main Campus)	Jaja	Male	660
	Mariere	Male	444
	Moremi	Female	866
B (New Hall)	Eni Njoku	Male	800
	Aliyu Makama Bida	Female	764
	Fagunwa	Female	276
	Madam Tinubu	Female	524
	Sodeinde	Male	968
C (Gate/Education)	El Kanemi	Male	526
	Kofo Ademola	Female	512
	Queen Amina	Female	646
	Saburi Biobaku	Male	512

Table 2 – Specified Halls for certain categories

CATEGORY	SPECIFIED HALLS	
	MALE	FEMALE
Students with peculiar health	Jaja	Moremi
Scholars	Mariere	Moremi
Sports men and women	El-Kanemi	Queen Amina

The objective of the HSAP include the satisfaction of the following soft constraints: 1) as many final year students as possible are accommodated; and 2) as many scholars as possible are accommodated. The allocation of final year category takes pre-eminence over all other allocations with the exception of health, sport and foreign categories. The quality of an allocation (solution) in HSAP can be measured in terms of 1) satisfaction and non-violation of given hard constraints, 2) space utilization, i.e. the amount of space that is wasted and the amount of space that is overused (allocation exceeding the capacity of a given hall), and 3) extent of satisfaction of given soft constraints. Though desirable, an optimal solution is not always achievable with the application of heuristics (Landa 2003).

3 Genetic Algorithm

Computer simulation of GA makes a population of abstract representations (chromosomes) of candidate solutions (individuals) of an optimization problem evolve toward better solutions based on it parametric instances of mutation and crossover. Since the allocations at hall and block/floor levels are mutually exclusive, the algorithm is applied in a slightly different multi-level fashion with two similar chromosome representations for each of these stages of the allocation process. Hall allocation has the structure $(c_1\ (h_{11} \dots h_{1n}),\ c_2\ (h_{21} \dots h_{2n}) \dots c_m\ (h_{m1} \dots h_{mn}))$ where c_i represents a category i, h_{ij} represents allocation of category i in hall j, m is the number of categories and n is the number of halls. The population consists of N individuals, where N is the population size. For block and floor allocation, the chromosome structure has the form $(c_1\ (b_{11}\ (f_{111} \ldots f_{11x}) \ldots b_{1w}\ (f_{1w1} \ldots f_{1wx})) \ldots c_m\ (b_{m1}\ (f_{m11} \ldots f_{m1y}) \ldots b_{mw}\ (f_{mw1} \ldots f_{mwy})))$ where c_i represents a category *i*, b_{ij} represents block *j* in category *i*'s allocation, f_{ijk} represents category *i*'s allocation on floor *k* of block *j*, *m* is the number of categories *w* is the number of blocks and *x* and *y* are the number of floors on the respective blocks.

3.1 GA Operations

In a bid to promote a well-spread search of the space solution, we allowed the initial generation to be generated randomly in a "greedy" fashion at each level of the allocation. Roulette wheel selection scheme was used to select individuals through a fitness-based process with better solutions having better chances of been selected. A one-point crossover strategy was used for recombination of selected parents while a repair algorithm was invoked to ensure that the offspring produced do not exceed the capacity of the hostel (in case of hall allocation) or the capacity of the floor (in case of block and floor allocation). Mutation was carried out based on a specified mutation rate and a randomly generated variable. At the hall level, two halls and two categories are selected randomly. A random number, *q*, is generated, where $q\ \varepsilon\ [1, p]$ and *p* is the smaller of the allocations for the first category in the first hall and the second category in the second hall. The allocation of the first category in the first hall is increased by *q* while the allocation of the second category in the same hall is decreased by *q*. The reverse is performed for the second hall to ensure that the hall capacities are not exceeded. The same process is repeated for block/floor allocation, substituting of floors for halls. The algorithm terminates if any of the following three conditions are met: 1) an optimal solution is found, 2) for fifty consecutive generations, the average fitness of the current population is at least 95% that of the previous one and 80% of the best fitness found throughout execution, 3) the execution has been carried out over a specified number of generation and conditions (1) or (2) are not satisfied.

The fitness of an individual in the population is measure in term of the satisfaction of given constraints. Weights are assigned accordingly to the constraints that impose limitation on how spaces can be allocated at both hall and block/floor levels. The set of fitness values is represented as a 1-dimensional array of real numbers between 0 and 1 (inclusive). A value of 0 indicates complete violation of all given constraints while a value of 1 indicates non-violation of all constraints hence an optimal solution.

4 Experimental tests

We conducted series of simulation experiments based on given data set with the aim of finding combinations of the parameters such as population size (*N)*, crossover rate (P_c) and mutation rate (P_μ) that consistently affect the performance of the algorithm and give optimal (or good sub-optimal) results. We studied the effect of P_c ε [0.1,1.0]. We used 10 different values of P_c in [0.1,1.0] with an increment of 0.1. For each fixed P_c, we then conducted 5 runs each for P_μ ε [0.0, 0.1] with an increment of 0.01. We then take the average. Hence there are 10 average fitness values. The same process is repeated to see the effect of P_μ in two different intervals e.g. [0.0, 0.1] and [0.1,0.9]. In this experiment, P_c was varied from 0.1 to 1.0. To determine the convergence of results obtained, we compared the best fitness and average fitness values obtained over a number of generation runs. As the number of generations move away from forty-five, the best fitness and average fitness values were seen to converge (Fig. 1). The best effect of crossover rate on the performance of the algorithm was observed with P_c in the range 70-80% with some good results also at 10-20%, 40-50% (Fig. 2). Values of P_c that is higher than 80% seems to worsen the performance of the algorithm. Similarly, we experimented separately with two different set of P_μ (0-0.1 and 0.1-0.9) (Fig. 3&4) to be able to know which set of P_μ will give the best optimal solution. A linear regression was plotted for each of the two set of P_μ. Although the algorithm improved in performance with the increase in value of both sets of P_μ generally, it gives better performance when P_μ is kept between 0-0.1 (goodness of fit is 0.922 compared with 0.778 for P_μ values 0.1-0.9). Considering best fitness, good solutions can also be obtained with values of P_μ between 40-80%. The performance of the algorithm tends to increase with increase in population size for both male and female allocation although the rate of change for the female is slightly higher than the male (0.837 compared with 0.616) (Fig. 5&6). This might be explained with the fact that the number of female applicants and constraints related to this category are generally lesser than that of the male.

A study to determine the rate of feasibility of the solutions obtained during experimental runs was also carried out. A solution is feasible if it does not violate any of the hard constraints. For the study, the number of generations was fixed at 1000. The algorithm was made to execute 50 times independently. For each execution, the number of generations evolved, total number of feasible solutions over all generations and the feasibility rate (as percentage of the total number of solutions) were computed. The results obtained with $P_c = 0.3$, $P_\mu = 0.3$, and $N = 100$ is presented in form of scatter graphs in Fig. 11&12 for male and female allocations respectively. The scatter graphs reveals the random nature of GA (the rate of feasible solutions covers a very wide range and also conforms to Normal Distribution). The results show that feasibility rate improves with increasing number of generations.

5 Conclusions

HSAP is an interesting instance of the SAP where heuristic is just being applied to obtain an optimization solution. We have successfully applied GA to solve this problem. It has thus been shown that heuristics and metaheuristics techniques can be successfully applied to solve this class of problem. Our current work is exploring other heuristics and metaheuristics approaches with some innovative ideas in handling this problem.

Acknowledgement. The authors appreciate the support of the University of Lagos, Nigeria and the School of Computational and Applied Mathematics as well as the Faculty of Science of the University of Witwatersrand, South Africa towards the success of this work.

References

Adewumi AO, Ayeni JOA, Abass O, Fasina EP (2003). Some Comments on Task Allocation Problem in Distributed Computing Systems J. Comp. Sci. and Its Applications, Nigeria, 9(1), June, 136-145.

Adewumi AO, Ihemedu N, Ayeni JOA. (2005). An Evolutionary-based Approach to University Course Time-Tabling Problem. First Annual Research Conference and Fair, University of Lagos, October 26, No. 9.11

Bai R (2005). An Investigation of Novel Approaches for Optimising Retail Shelf Space Allocation. PhD Thesis. School of Computer Science and Information Technology, University of Nottingham, UK

Benjamin C, Ehie I, Omurtag Y (1992). Planning Facilities at the University of Missoury-Rolla. J. of Interfaces, 22(4), 95-105.

Burke EK, Varley DB (1998). Space Allocation: An Analysis of Higher Education Requirements. In: Burke, E.K. and Carter, M.W. (eds.) The Practice and Theory of Automated Timetabling II (PATAT 1997): Lecture Notes in Computer Science, 1408, Springer, 20-33.

Burke EK ,Varley DB (1998b). Automating Space Allocation in Higher Education. Selected Papers from the 2nd Asia Pacific Conference on Simulated Evolution and Learning SEAL 98, Lectures Notes in Artificial Intelligence, 1585, Springer, 66-73.

Giannikos J, El-Darzi E, Lees P (1995). An Integer Goal Programming Model to Allocate Offices to Staff in an Academic Institution. J Op Research Society, 46(6), 713-720.

Goldberg DE (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Addison Wesley.

Landa JD (2003). Metaheuristics and Multiobjective Approaches for Space Allocation. PhD Thesis. School of Computer Science and Information Technology, University of Nottingham, UK.

Ritzman L, Bradford J, Jacobs R (1980). A Multiple Objective Approach to Space Planning for Academic Facilities. J Management Science, 25(9), 895-906.

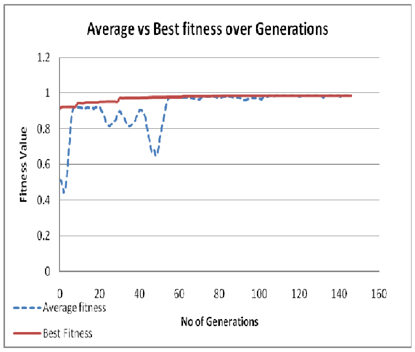

Fig. 1 Average versus Best Fitness over a number of Generations (N = 30, P_c = 0.1, P_μ = 0.2)

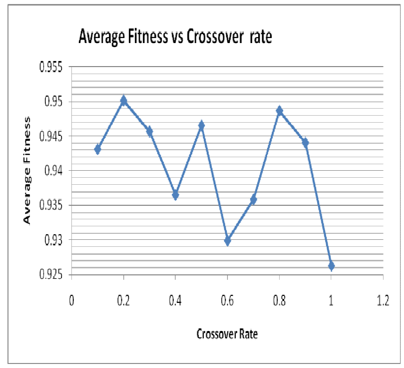

Fig. 2 Average fitness vs Crossover rate

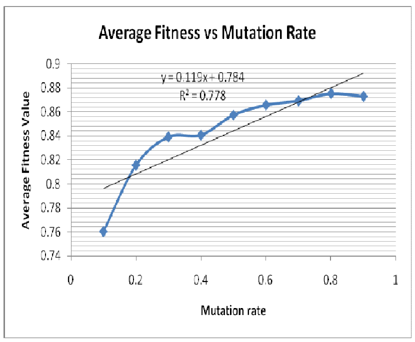

Fig. 3 Average Fitness vs Mutation Rate (0.1-0.9)

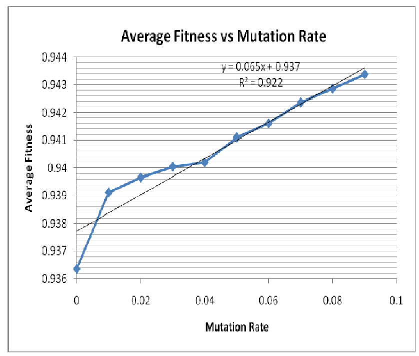

Fig. 4 Average Fitness vs Mutation Rate (0.0-0.1)

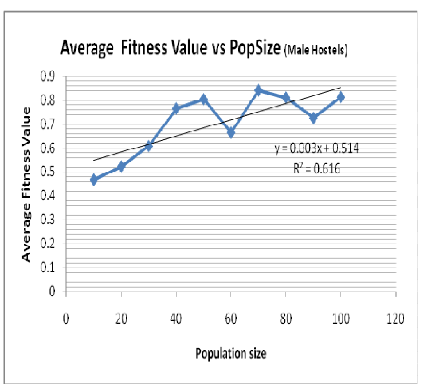

Fig. 5 Average fitness vs popsize (P_c = 0.1, P_μ =0.1) (Male)

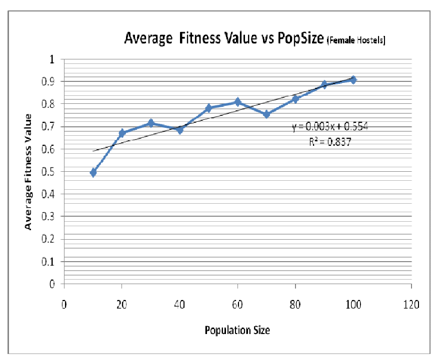

Fig. 6 Average fitness vs popsize (P_c= 0.1 P_μ=0.1) (Female)

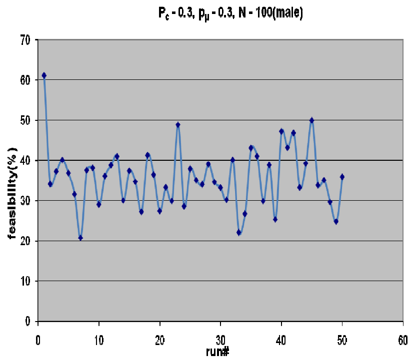

Fig. 7 Scatter graph of feasibility study on the combination (0.3, 0.3, 100) (male)

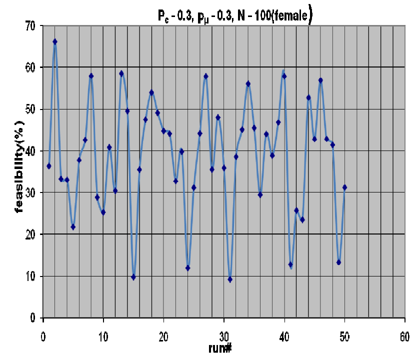

Fig. 8 Scatter graph of feasibility study on the combination (0.3, 0.3, 100) (female)