A Framework of Investigating the Impact of Different Data Densities on the Rates of the Genetic Operators in the Matrix-based EAs Model

Li-Pei Wong[1], A. T. Khader[2]

[1] Faculty of Information Science and Technology (FIST),
Multimedia University (Melaka Campus),
Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia.
lpwong@mmu.edu.my
[2] Artificial Intelligence Lab, School of Computer Sciences,
Universiti Sains Malaysia, 11800 Minden, Pulau Pinang, Malaysia.
tajudin@cs.usm.my

Abstract. This paper intends to investigate the relationship between the data density and the search operator rates in the examination room assignment problem. For this, we have suggested the Matrix-based EAs Model to solve the examination room assignment problem. In this particular problem, we allocate examinations into a number of rooms for a particular slot. Thus, data density in this paper refers to the total number of candidates for all the examinations divided by the maximum capacity of all available rooms in a particular slot. We believe that for better performance, the probability values of search operators need to be adjusted inline with different data densities.

1 Introduction

Evolutionary Algorithms (EAs) is an umbrella phrase used to illustrate a computer-based problem solving systems, which utilizes a computational model of evolutionary processes as key fundamentals in their design and implementation. In these algorithms, encoding and reproduction mechanisms are used to solve some difficult problems based on the principle of evolution – survival of the fittest [1]. The search operators such as crossover, mutation and reproduction are applied to evolve the solutions based on certain probability values. However, it is usually very difficult to determine or estimate an optimal set of search operator rates for a problem.

The aim of this paper is to study the significance relationship between data density and the genetic operator rates in the examination room assignment problem. Let us denote by P_C as the probability of applying crossover, P_M as the probability of applying mutation as P_R as the probability of applying reproduction. We have applied the Matrix-based EAs Model [2] in solving the examination room assignment problem. In this particular model, we employ matrix as a basic unit in a population (chromosome). A chromosome is a candidate solution for the problem. The matrix consists of rows and columns. Each row represents a room whereas each column represents an examination in a particular time slot. The matrix is assigned with a set of numbers. These numbers represent the total candidates allocated in a room. If we divide the total number of candidates for all the examinations by the maximum capacity of all available rooms in a particular slot, we will acquire the data density of that particular slot. We believe the data density has a significant impact on the rate of genetic operators.

2 Examination Room Assignment Problem

Examination room assignment problem is an instance of resource allocation problem. This problem is a NP-complete problem [3], in which events (examinations or subjects) have to be arranged into a number of rooms/venues, subjected to a set of constraints. There are two basic entities in this problem: examination and room. Hence, our aim is to allocate examinations into a number of rooms for a particular slot. The allocation plan also must fulfill some of the hard and soft constraints.

In this particular problem, two possible relationships may exist between examination and room: one-to-one relationship and many-to-many relationship. In the one-to-one relationship, a room may accommodate only one examination. Sometimes, it is difficult to accomplish this requirement since there is a possibility where an academic institution lacks of any large examination hall that is able to accommodate all the candidates. Thus, we need to split a large examination to several rooms or allow a room to be shared by several examinations. This situation is recognized as a many-to-many relationship. Considering all these factors simultaneously in producing a feasible allocation plan is a difficult task. We may acquire more information about university examination timetabling problem via [4]. It provides a comprehensive survey regarding the university examination timetabling in Britain.

3 Matrix-Based EAs Model

In this Matrix-based EAs Model, We adapt matrix as the representation scheme. The matrix consists of rows and columns. Each row corresponds to a room whereas each column corresponds an examination in a particular time slot. The matrix is assigned with the number of candidates allocated in a room. Refer to Fig. 1 for a conceptual representation of a chromosome. That figure illustrates the examination with code AKW101 is split over R1 and R4. At the same time, R1 accommodates candidates that take the examination AKW101, AKW102 and MAA102.

For the algorithm part, we adapt the algorithm that is stated in Koza [5] (See Fig. 2). We create a population with a number of chromosomes. These chromosomes are then evaluated by a fitness function. We perform a summation technique to design the fitness function and it consists of four different evaluations. Each evaluation in the fitness function corresponds to a soft constraint that needs to be optimized. This fitness function eventually gives a penalty value as the final outcome. Based on the penalty value, we select the promising chromosomes for further transformation. For this, we employ the Roulette Wheel selection. The selected chromosomes will either be applied reproduction, crossover or mutation. Only one operator is chosen at a time. In other words, $P_R + P_C + P_M = 1$. The transformed chromosomes are then inserted to the new population. We repeat these steps until the individuals in the new population reach a limit. This is a complete cycle for one generation. We usually reiterate the system until designated result is obtained or the fitness value converges to a point.

Courses

Rooms	AKW101	AKW102	CAT102	CAS102	MAA102	HTU201
R1	70	60	0	0	30	0
R2	0	20	0	0	0	0
R3	0	20	0	0	0	60
R4	80	0	0	0	0	0
R5	0	0	150	0	0	0
R6	0	0	0	200	0	60
R7	0	0	0	0	450	60

Fig. 1. Conceptual Representation of a Chromosome

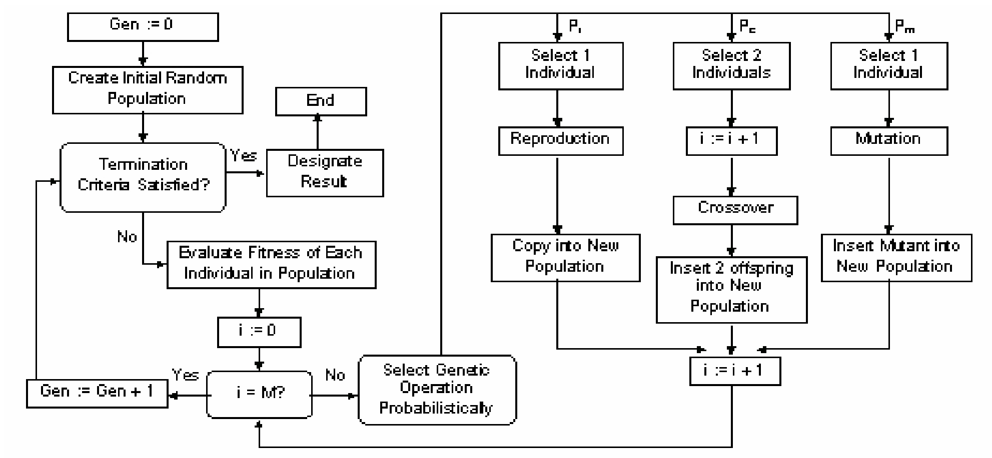

Fig. 2. Flowchart of Evolutionary Algorithms

4 Data Densities versus Search Operator Rates

In the Matrix-based EAs Model, we assign the matrix with a set of numbers. These numbers represent the total candidates that sit for an examination in a particular room. We may have several examinations in a particular slot. For instance, we may have ten examinations to be conducted in a slot. Let's assume there are a total of 3500 candidates sitting for those ten examinations. Also, let's assume the capacity of available rooms is 4000. With these two pieces of information, we emerge the data density for that slot as total candidate per room capacity, which is 3500 / 4000 = 0.875. We believe that the data density has a great impact on the setting of search operator rate in the Matrix-based EAs Model.

Also, in order to study the impact of different data densities on the rates of genetic operators, we assume that having four types of scenarios; each of them characterizes a slot with different data densities, which is high, intermediate, low and extremely low. The room capacity remains unchanged for those four slots.

For a slot with a high data density, we anticipate a high crossover rate (P_C) will produce individuals with relatively high penalty point. This indirectly decreases the overall fitness of a population. On the other hand, we believe that a high reproduction rate (P_R) and mutation rate (P_M) will produce satisfactory results. This makes sense, as the slot with high data density will only have limited empty seats to perform optimization. For instance, assume that we have 2500 empty seats and there are 2400 candidates to be allocated. The data density for that slot is 0.96, which is quite high. Thus, the system will only has 2500 – 2400 = 100 empty seats to perform optimization. This similar setting (low P_C, high P_R and P_M) also applies to a slot with extremely low data density. We believe that the slot with limited empty seats will require relatively high P_R and P_M to fine-tune the results. High P_C tends to disrupt the system.

For a slot with an intermediate data density, we expected a high P_C would help in improving the overall fitness of a population. P_R and P_M should be decreased. This is because a slot with intermediate data density will has adequate empty seats to perform optimization. For instance, assume that we have 2500 empty seats and there are 1800 candidates to be allocated. The data density for that slot is 0.75, which is relatively moderate. Thus, the system will has 2500 – 1800 = 700 empty seats to perform optimization. This similar setting (low P_R and P_M, high P_C) also applies to a slot with low data density.

5 Experiment Design

In order to study the impact of the different data densities on the search operator rates, we propose to select four groups of data taken from Universiti Sains Malaysia[1] (USM). These four groups of data represent slots, which with high, intermediate, low and extremely low data density respectively.

On the other hand, we propose to group the experiments into three major categories. Each category examines the behavior of search operators with different probability value. Also, there are three experiments included in each category. Each experiment will have different setting of the parameter. For instance, in Category 1 that examines the behavior of reproduction:

- First experiment – the reproduction is prohibited to perform whereas the other two operators have equally remaining chance to perform.
- Second experiment – the reproduction has half the chance to perform whereas the other two operators have equally remaining chance to perform.

[1] Examination data for Semester 1 Academic Session 2001 / 2002.

- Third experiment – the reproduction has highly dominated the chance to perform whereas the other two operators have equally remaining chance to perform.

Similarly, the same pattern of parameters setting will be applied to Category 2 and Category 3. Those two categories are meant for examining the behavior of crossover and mutation. The details of the experiment design are shown in Table 1.

Table 1. Experiment Design Details

Category	Experiment	P_R	P_C	P_M
Category 1	Exp 1	0.00	0.50	0.50
	Exp 2	0.50	0.25	0.25
	Exp 3	0.90	0.05	0.05
Category 2	Exp 4	0.50	0.00	0.50
	Exp 5	0.25	0.50	0.25
	Exp 6	0.05	0.90	0.05
Category 3	Exp 7	0.50	0.50	0.00
	Exp 8	0.25	0.25	0.50
	Exp 9	0.05	0.05	0.90

6 Conclusion

We believe that this study will aid us in studying the behavior of reproduction, crossover and mutation operator with different data densities. The results of the experiments will then to be translated into graphs type of average penalty value versus generation.

Recently, more and more researchers are putting their focus in estimating or determining the parameters for the Evolutionary Algorithms. Hence, we hope that this piece of work would contribute some ideas to them.

References

1. Davis, L.: HANDBOOK OF GENETIC ALGORITHMS. Van Nostrand Reinhold, Chichester, New York (1991)
2. Wong, Li-Pei. and Khader, A.T.: Solving Examination Room Allocation Problem by Evolutionary Algorithm. In: Proceeding of the International Conference on Operations Research for Development, Chennai, India (2002)
3. Garey, M. R. and Johnson D. S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman Co., San Francisco (1979)
4. Burke, E. K., Elliman, D. G., Ford, P. and Weare, R. F.: Examination Timetabling in British Universities - A Survey. In: Burke, E. K., Corne, D., Paechter, B. and Ross, P. (eds.): PATAT '95 Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling, (1995) 423–434
5. Koza, J. R.: Genetic Programming On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge, Massachusettes, London, England (1992) 29