A Multi-Objective Genetic Algorithm for Solving Conflicted Goals in Questions Generating Problems

Nur Suhailayani Suhaimi
Department of Information System
Faculty of Computer Science and Mathematics
Universiti Teknologi MARA,
Shah Alam, 40450 Selangor, Malaysia
suhailayani@tmsk.uitm.edu.my

Siti Nur Kamaliah Kamarudin
Department of Information System
Faculty of Computer Science and Mathematics
Universiti Teknologi MARA,
Shah Alam, 40450 Selangor, Malaysia
snkamaliah@tmsk.uitm.edu.my

Zalinda Othman
Department of Information Technology
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia,
Bangi, 40450 Selangor, Malaysia
zalinda@ftsm.ukm.my

Norazam Arbin
Department of Mathematics
Faculty of Computer Science and Mathematics
Universiti Teknologi MARA,
Seri Iskandar, 32610 Perak, Malaysia
noraz647@perak.uitm.edu.my

Abstract — **Multi-objective genetic algorithm (MOGA) has been used for more than a decade to solve real-world optimization problems that have several, and often conflicting objectives. In this research, the conflicting objectives of achieving the maximum accuracy of the solution and at the same time minimizing the redundancy of the optimal solutions in retrieving the best set of exam questions for academicians for a particular subject are highlighted. Hence, the aim of this paper is to solve the multi-objective problem in a chromosome (solution) and also to maintain the fitness of the chromosome. The results of this research are measured based on the similarity achieved between the obtained and desired solutions. By using MOGA, a promising result is obtained with the maximum accuracy and simultaneously, minimizing the redundancy of the genes in a solution.**

Keywords - component; multi-objective, MOGA, genetic algorithms

I. Introduction

Multiple objective problems are usually problems that have a set of optimal solutions where unlike single objective problems which usually have a defined optimal solution [1]. One way to deal with the multi-objectives problems is by using Multi-Objectives Evolutionary Algorithm (MOEA). Evolutionary Algorithms (EAs) have several advantages which are why researchers opted to use optimization techniques to solve problems that are too complex for gradient method or linear programming. Gradient methods or linear programming is sufficient enough to find the optimal solution for the single objective optimization problems. However, real world problems are often complex where researchers will have to solve several conflicting objectives concurrently. For example, finance, engineering and economics are some of the research areas which deal with multi-objectives problems.

In general, EA use the concept of Darwinian evolution of survival of the fittest. Using the EA terminologies, candidate solutions (chromosomes) are grouped in one population where parents' selection process is then made according to their fitness values to create offspring. The fitness of the offspring will then be compared to the parents and other chromosomes to be included in the population of the next generation. Theoretically, the generations' fitness values will be improved with each runs. One of the techniques in EA is Genetic Algorithm (GA). GA has the advantage of being robust and the operations of crossover and mutation in GA bring diversity to the offspring produced. Due to the GA's advantages, researchers have opted to use GA to solve the multi-objectives problems in their research. Multi-objectives genetic algorithm (MOGA) was first introduced over a decade ago and have been expanding ever since [2] with promising results achieved by various researchers.

The ability of MOGA to solve complex multi-objectives problems is a motivation for this research. Following this, the paper is organized as such. Section II discusses the previous researches which uses MOGA as the main technique, followed by Section III with the methodology. Section IV is the research's result and Section V concludes the research conducted.

II. Literature Review

Optimization often refers to as a way to solve a problem by finding the best possible solution while taking into consideration of the limit presented in the problem [12]. However, most real-world problems usually, not only consists of more than one objectives, but also having several limitations that decision makers will have to consider to in attempt to solve the problem ([8], [9], [12]).

Over the years, researchers have been using MOEA in research areas such as engineering [3, 4, and 5], transportation [6] and pattern recognition [7]. Early studies involving MOEA often uses test problems that were relatively simple and not scalable or problems that were too complex for the researchers to visualize their Pareto-optimal front result [8]. However, [8] further claimed that the problem was solved in 1999 with the introduction of a systematic procedure to create a simple and scalable test problems, which subsequently allows future researches to further solve multiple-objectives optimization problems. [8] and [9] added that since most real-world decision problems naturally have more than one objective, this subsequently makes the multi-objective optimization problems an important area of research. With the growing number of research using MOEA and the unanswered questions in this field, [9] remarked that there are no universal guideline to the definition of "optimum" in an MOP research where most of the time, it is up to the researcher themselves to determine the word's definition. The claim supports the need for a comparative study on the MOEAs' abilities in solving multi-objective problems ([8], [9]).

However, in order for a comparison study to be conducted on MOEAs, it is important to first look at the basis for an MOEA study. The fundamental of both single objectives and multi-objectives optimization, including MOGA, are based on the natural process of biological evolution [4]. Subsequently, as compared to single objectives optimization problems, MOPs are essentially more difficult to solve. One research highlighted the difficulties in using multi-objectives optimization which includes the problem of assigning fitness function and selection process while at the same time preventing premature convergence by maintaining the diverse population with a well distributed trade-off front [4]. For example, the difficulties in using MOGA also include high calculation costs as numerous iterations are needed to calculate the values of both the constraints and objective functions [3]. In addition, the absence of a generally well-accepted root method to solve a variety of multi-objective optimization problems has also become a basis for future works to be done to solve optimization problems with more than one objectives [9].

However, the difficulties in using EA techniques to solve multi-objectives problems, including MOGA, did not deter the researchers from using the techniques itself. [3] used MOGA as part of their model called Divided Range Multi-Objective Genetic Algorithm (DRMOGA) for parallel processing. The aim of DRMOGA was to efficiently find Pareto solutions by testing it to four test functions on PC cluster systems as compared to single population model and distributed model [3]. Parallel processing was also noted in the research by [5] where MOGA were used to optimize the batch plant design in the chemical engineering area of research. Using a simple mutation and crossover procedures with a binary encoding, the researchers created an optimization framework to deal with the multi-objectives problem. Their version of MOGA was first tested against several test performances before testing it against two case studies in the batch plant design problems [5].

Solutions in a multi-objective optimization problem are usually represented as a set of optimal answers in a search space, provided that the set best fulfills all objectives and no other solutions are superior to that particular set [9]. Besides that, the set of solutions will consider to be a Pareto-optimal solutions if no other solutions in the search space will inadvertently decline some conditions and not cause other conditions to concurrently grow in at least one other condition ([12]). The set of optimal answers or solutions are also known as Pareto-optimal solutions, where the solutions provide flexibility to assists decision makers [9]. The Pareto-optimal solutions representation has been seen to be applied in many multi-objective optimization problems, including MOGA ([4], [6], [7], [10]). [4] used the Strength Pareto approach with EA (SPEA) to optimize the shape of electro kinetic micro channels. According to [4], the use of SPEA was easier to implement as compared to gradient based methods which involved mathematical calculations that are too complex to be solved in the fluid dynamical problems.

In transportation problem, [6] aimed at iteratively updating the population based on clustering algorithm while maintaining a finite-sized archive of the Pareto-optimal solutions. By integrating the local search technique with GA, the quality of the solutions achieved in solving the multi-objectives transportation problem (MOTP) was improved [6]. Besides clustering, MOGA has also been used previously for optimizing classification systems in pattern recognition area of research [7]. MOGA was applied as part of the two optimization process in the research. The first process was to optimize the feature extraction of handwritten digits and uppercase letters technique called Intelligent Feature Extraction (IFE). Results from the IFE was later used for optimizing the ensemble of classifiers (EoC) in which another type of MOGA was integrated into the technique [7].

Multi-objective optimization are also applied in the engineering field ([10], [11]). [10] aimed to optimize two conflicting objectives for their research where the first is to achieve the optimal removal rate for productivity process and the second conflicting objective is to optimize the delamination factor which represents the superficial quality of the composite material. For this research, a micro-genetic algorithm technique was also employed as part of the optimization process while a Pareto's front was used to represent the multiple optimal solutions for the decision making process [10]. Meanwhile, [11] focused on optimizing the machine processing time and while at the same time, fulfilling the technological and material limitations in their research. GA was also chosen to be the attribute optimizer to solve the problem of automating the feature selection for metal cutting in their machine. [11] highlighted that among the advantages of using GA is the capability of solving multi-object optimization where for their research, as demonstrated by other related works in [3] ,[4] and [5]. Ultimately, the aim of these researchers is mostly to obtain a set of Pareto optimal solutions that satisfies the conditions of their respective research.

III. METHODOLOGY

The MOGA used in this study is developed by using the basic concept of conventional Genetic Algorithm. MOGA also have the basic phases of Population, Selection, Crossover and Mutation. The multi-objective feature is embedded in the Crossover operator.

Concerning the conflicted goals in generating the questions set, the chromosomes are adjusted into two sections. The goals are conflicted with each other whereas we want to maximize the accuracy of retrieving questions desired and minimizing the redundancy of genes.

A. Chromosome Representation

The genes in a chromosome were arranged in a length of forty. The first twenty genes sorted according to dissimilarity of genes. Genes who are similar or almost similar to each other are sorted in the last twenty genes as shown in the Fig. 1.

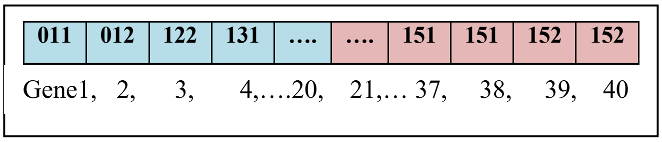

Fig. 1. Example of a chromosome representation

B. Population

An initial population is generated randomly. Five candidate solutions initially retrieved from database by calculating the distance measure of the chromosome with desired solution entered by user. Five most similar chromosomes were selected as candidates to undergo Genetic Algorithm operators.

$$\sum_{n=40}^{i=1} d\left(\left|x_i - y_i\right|\right) + d\left(\left|x_{i+1} - y_{i+1}\right|\right) + \ldots.. + d\left(\left|x_n - y_n\right|\right) \quad (1)$$

Where n denotes the length of chromosomes, x denotes the chromosome in database, y denotes the desired chromosome. i denotes the genes. The summation of genes value will be the similarity value of the chromosome.

C. Fitness Function and Parent Selection

Fitness function is count in two criteria. First criterion is maximum value of matched genes between expected solution with desired. The second criterion is minimum number of redundant genes. To ensure the chromosome fitness is standardized, we took the value of unmatched genes for the first criterion. Minimum value of unmatched genes and minimum redundant genes considered as fittest value

$$fs_{\min} = X_i + X_{i+1} + \ldots.. + X_n \quad (2)$$

where $fs_{\min}$ denotes the fittest chromosome, X represent the chromosome, i denotes the gene number and n denotes the length of the chromosome.

D. Crossover

Two fittest chromosomes that have selected as parents will undergo the crossover operator. A 3-point crossover take placed after parent selection. First cut-off would be between 20^{th}-gene and 21^{st}-gene, in order to separate the chromosome into two criteria. Second cut-off randomly occurred within the first 20 genes and the third cut-off randomly within the last 20 genes as shown in Fig. 2.

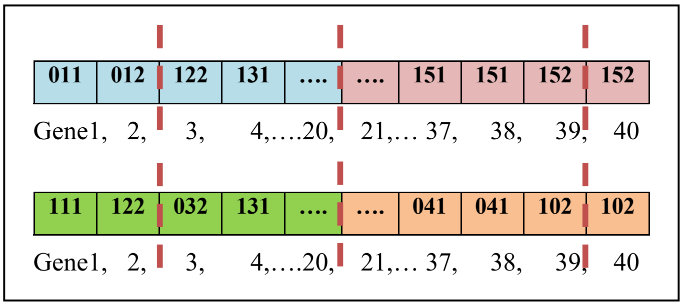

Fig. 2. Crossover operator

E. Mutation

The new chromosomes obtained from crossover operator will then undergo the mutation to maintain the fitness of the chromosome between the ranges of R_1. This is due to avoid the chromosome diversity far between its stability. After the fitness is calculated, the mutation operator will only apply if the fitness falls out of the ranges. Eq. (3) shows the range, R_1 which every offspring produced must fulfilled after crossover operator. The value of p denotes the minimum range and q is the maximum range value. The mutation will only take place if the range is fulfilled as shown in Fig. 3.

$$R_1 = p \leq fs \leq q \quad (3)$$

The mutation probability was set lower than 0.05 per cent to reduce the chromosome being mutated frequently for each generation. Mutating too much could lead to the optimization falls out of their population pool and produce worst offspring.

DOI 10.5013/IJSSST.a.15.03.05

ISSN: 1473-804x online, 1473-8031 print

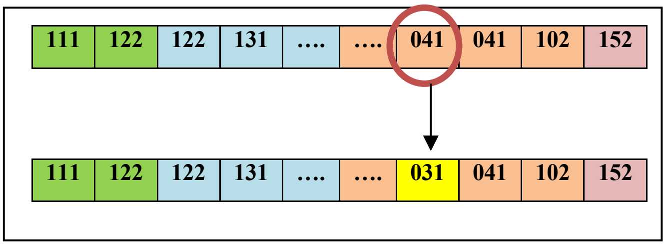

Fig. 3. Mutation Operator

F. *Stoping condition*

The iteration of the generations will be halted if the desired chromosome obtained at maximum fitness of 80% or the iteration number reached 100 generations. The MOGA is tested 100th times for every generation. The best chromosome produced after 10th generation for each test is recorded.

IV. RESULT ANALYSIS

TABLE I. RESULT A

GENERATION	BEST FITNESS (%)	AVERAGE FITNESS (%)
10	42.73	38.12
20	49.69	41.76
30	56.14	44.81
40	58.36	50.28
50	58.21	53.17
60	63.16	55.42
70	63.75	57.35
80	69.54	60.79
90	70.14	61.43
100	72.32	64.98

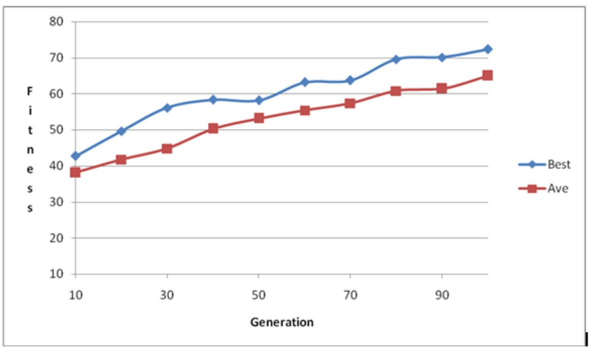

Fig. 4. Best and Average Fitness

Fig. 4 shows the best fitness for every ten (10) generations up to hundredth (100th) generations. Clearly displays that the fitness's are increasing as the number of iteration increased. The average accuracy of every 10 iterations gathered to be further analyzed. Begin with 1st generation, the fitness rose rapidly up to 40 per cent at the generation 10th. The fitness produced after another ten generation slowly increased to approximately 50 per cent.

Each generation will produce better chromosome fitness mapped to the target. However as human genetic reproduction, there's a time where the fitness decreased, our MOGA also facing the same situation as can be seen in the 40th generation to 50th generation. The best fitness decreased 0.15 per cent from 40th generation to 50th generation.

At the 100th generation, the best fitness produced after MOGA implemented is above 70 per cent. Due to maintaining the fitness of the chromosomes from falls far from their stability, the iteration of MOGA halted at 100th generation. This is how the second objective of this study achieved.

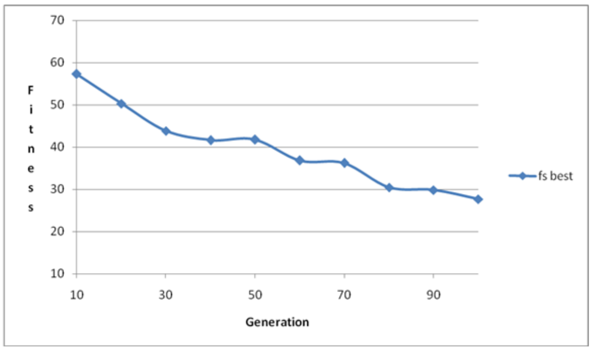

Fig. 5. Distance between Best Fitness and Target

As shown in Fig. 5, the graph is representing the distance between best fitness chromosome and desired chromosome (target). As the generation of MOGA increased, the distance is closer to the target. At the 100th generation, the gap is only approximately below 30 per cent to the target solution. Therefore, it is proven that MOGA is suitable to solve the conflicted goals in this study.

Table II shows the fitness value for the mutation probability 0.5, 0.05 and 0.01. The test was conducted in order to observe the changes of fitness value over 100 generation. The tests begin with the mutation probability of 0.5 which increased the tendency for the chromosome to undergo the mutation process. In order to observe the increment or decrement of the fitness value, the value of mutation probability being decreased to 0.05 and 0.01 for each 100 generation.

TABLE II. RESULT B

GEN	P_{mut} <= 0.5	P_{mut} <= 0.05	P_{mut} <= 0.01
10	44.60	38.12	31.02
20	48.51	41.76	31.02
30	55.63	44.81	33.46
40	68.11	50.28	33.46
50	81.02	53.17	33.46
60	88.36	55.42	33.46
70	94.13	57.35	38.87
80	100.00	60.79	38.88
90	halt	61.43	38.88
100	halt	64.98	39.71

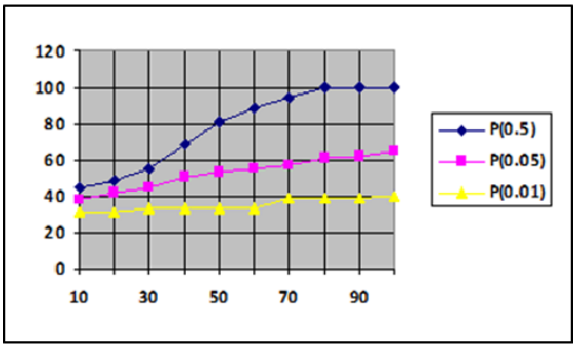

Fig. 6. Fitness Value with various mutation probabilities

The average of fitness for each iteration shown in Figure 6 indicating the increment of the fitness value over the 100 generation in every running. By the hypothesis of as the probability of mutation increased, the tendency of the chromosome to mutate also increased and lead to the increment of the fitness value.

Based on the graph shown, the probability of mutation set will affect the fitness value over the GA generation. Rapid increment in the fitness value occurred when the probability is set to 0.5. The value increased tremendously and stuck at the local optima. Thus, the probability reduced to 0.05 and the changes just fit to the chromosome normal range.

The probability of mutation also tested to 0.01 in order to test whether it will produce better increment. However the 0.05 probability seems better due to the changes in 0.01 probability rarely happened. The fitness barely increased due to the probability near to zero (0).

V. CONCLUSION AND FUTURE WORKS

Two goals are aim to be achieved in this research study. The first goal is to maximize the accuracy of solution retrieved. The second goal is to minimize the redundancy of the genes in a solution. In order to complete the task, MOGA is applied. MOGA is aim to solve real-world problem which is effective enough in solving conflicted goals problem.

The optimization problems often can be solved using GA where there is no conflict in the objectives. However, the GA as an optimizer was usually used as one-way optimization whether to maximize or to minimize. Therefore MOGA is applied in this study.

MOGA is sufficient enough to solve the multi-objectives problem and eventually decreased the gap distance to the target. However, the GA problem of stuck in local optima remains. The generation of MOGA limited at 100th generation to maintain the stability of the chromosome. In future, MOGA can be enhanced to reduce the local optimal problem by association with other AI techniques.

REFERENCES

[1] P. D. Justensen, Multi-Objective Optimization using Evolutionary Algorithms, Department of Computer Science University of Aarhus, 2009.

[2] D. A. Van Veldhuuizen and G. B. Lamont, Multiobjective Evolutionary Algorithm Research: A History and Analysis, Department of Electrical and Computer Engineering Air Force Institute of Technology, 1998.

[3] T. Hiroyasu, M. Miki and S. Watanabe, The New Model of Parallel Genetic Algorithm in Multi-Objective Optimization Problems – Divided Range Multi-Objective Genetic Algorithm, .

[4] I. F. Sbalzarini, S. Mullert and P. Koumoutsakos, Multiobjective Optimization using Evolutionary Algorithms, Center for Turbulance Research, Proceedings of the Summer Program 2000, pp. 63-74.

[5] A. Dietz, C. Azzaro-Pantel, L. Piboulea and S. Domenech, Strategies fro Multiobjective Geentic Algorithm Development: Application to Optimal Batch Plant Design in Process Systems Engineering, .

[6] S. A. Zaki, A. A. A. Mousa, H. M. Geneedi and A. Y. Elmekawy, Efficient Multiobjective Genetic Algorithm for Solving Transportation, Assignment and Transhipment Problems, Journal of Applied Mathematics, Vol 3, pp. 92-99, 2012.

[7] P. V, W, Radtke, R. Sabourin and T. Wong, Classification System Optimization with Multi-Objective Genetic Algorithm.

[8] K. Deb, L. Thiele, M. Laumanns, & E. Zitzler . Scalable multi-objective optimization test problems: Proceedings of the 4–9. Retrieved from http://repository.ias.ac.in/81671/, 2002.

[9] H. A. Abbass, R. Sarker, & C. Newton, C. PDE : A Pareto – frontier Differential Evolution Approach for Multi-objective Optimization Problems, 1999.

[10] R. Q. Sardiñas, P. Reis, & J. P. Davim, Multi-objective optimization of cutting parameters for drilling laminate composite materials by using genetic algorithms : Composites Science and Technology, 66(15), 3083–3088, 2006.

[11] D. M. D'Addona, & R. Teti, Genetic Algorithm-based Optimization of Cutting Parameters in Turning Processes : Procedia CIRP, 7, 323–328. doi:10.1016/j.procir.2013.05.055, 2013.

[12] C. A. C. Coello, Twenty Years of Evolutionary Multi-Objective Optimization : A Historical View of the Field, 1–20, 2005.