A Multi-Objective Bisexual Reproduction Genetic Algorithm for Computer Network Design

Sunantha Sodsee  
Department of Information Technology  
Faculty of Information Technology  
King Mongkut’s Institute of Technology North Bangkok  
1815 Pibulsongkarm Rd., Bangsue, Bangkok 10800 Thailand  
sunantha_s@hotmail.com, pym@kmitnb.ac.th

Phayung Meesad  
Department of Teacher Training in Electrical Engineering  
Faculty of Technical Education  
King Mongkut’s Institute of Technology North Bangkok  
1815 Pibulsongkarm Rd., Bangsue, Bangkok 10800 Thailand  

Abstract

This research is concerned with a genetic algorithm focusing on the Pareto based approach for solving multi-objective optimization problems. Multi-Objective Bisexual Reproduction Genetic Algorithm (MOBRGA) is proposed herein. MOBRGA uses a concept of sexual selection with different types and mutation rates in reproducing offspring. The comparisons among MOBRGA and other algorithms were performed for evaluation purposes. The results showed that the proposed algorithm performed well in some benchmark multi-objective optimization problems. In Shaffer’s and Murata’s problems, MOBRGA could find the Pareto front which is the line of the non-dominated solutions to the problems. The proposed algorithm was also tested with a computer network design, which is a multi-objective optimization problem. MOBRGA could successfully find network design solutions that meet user’s requirements.

Keywords: genetic algorithm, multi-objective optimization problem, Pareto based approach, sexual selection, non-dominated solution, MOBRGA, computer network design

I. Introduction

An optimization process consists of three basic things: an objective function, variables, and constraints [1]. The optimization process finds the value of the variables that minimize or maximize the objective function while satisfying the constraints. In the real world, it is unusual to see a problem that is considered only one variable which has only one minimum value. It means that when finding the solution, various combinations of the searching variables have to be considered in order to see which combination minimizes or maximize the objective functions [2-3]. A computer network design is an example. We must simultaneously consider many criteria such as cost, capacity, performance, and reliability of the network [4]. It is very difficult to decide which selection is the most suitable for the criteria because the criteria may conflict with each other. This dilemma is so called the multi-objective optimization problem (MOOP) [5].

There are two major approaches to solve the MOOP [2-3]. The first one which is a conventional technique is the linear combination of the different criteria in which different weights are assigned. The linear combination has two important drawbacks: all the solutions may not be found and weights assigned to some criteria may not be suitable. The second approach is an evolutionary algorithm (EA) which is a more suitable optimization technique. An EA is an artificial evolution process. It is an algorithm using the evolutionary technique such as natural selection and recombination to find an optimal configuration for a specific problem within specific constraints. EAs consist of the evolution strategy (ES), evolutionary programming (EP), genetic programming (GP), and genetic algorithms (GAs). A GA is a kind of multipoint search that has the advantages of optimizing the problem with multiple objectives and taking the similarity available in the family of possible solutions to the problem [5-6].

The multi-objective genetic algorithm (MOGA) [7-8] is an effective GA used to solve MOOPs. MOGA considers several objectives simultaneously. If the number of objectives increases then the complexity of problem also increases. This affects the solution, because the considered objectives often conflict with each other and the optimal solution to the problem is not obtained as a single perfect solution. It is a set of alternative solutions. There are many types of MOGAs. They employ different techniques in finding optimal solutions. Some examples of MOGAs are the plain aggregating approach, the population-based non-Pareto approach, the niche induction technique, and the Pareto-based approach [7].

The Pareto-based approach is the effective approach that is suitable for finding the possible solutions to MOOPs. It uses the principle of non-dominated solution and ranking selection approach for moving the possible solutions to the Pareto front which means that it can solve the MOOP and find the set of optimal solutions of the problem [7].

A computer network design is not a precise science. There are many good answers that involve trade-offs between cost and performance or reliability. It is very difficult for network designers to decide which solution is the most suitable for the criteria because the criteria may conflict with each other. GA can solve this problem.

This research intends to develop the Pareto-based approach of the multi-objective genetic algorithm for finding a set of the optimal solutions of the computer network design. It uses the concept of sexual selection and different mutation types of different sexes.

The remainder of this paper is organized as follows. Section II describes the literature review. Section III gives details of the proposed MOBRGA. Section IV illustrates the simulation results. Finally, Section V gives concluding remarks.
II. Literature Review

A. Multi-Objective Genetic Algorithm

A MOOP is a multi-criteria optimization, multi-performance or vector optimization problem [8]. It finds a vector of decision variables $x$, which optimizes a vector function satisfying inequality and equality constraints and whose elements represent the objective functions. These functions of performance criteria usually conflict with each other. The optimization seeks a solution which would give the values of all the objective functions acceptable to the decision maker.

Definition of a MOOP with “n” objectives: In MOOPs, we need to minimize or maximize the function

$$y = f(x)$$  \hspace{1cm} (1)

where $f(x) = (f_1(x), f_2(x), ..., f_n(x)); x = (x_1, x_2, ..., x_m) \in X; y = (y_1, y_2, ..., y_n) \in Y; x$ is a decision vector; $X$ represents parameter space; $Y$ represents objective space; and $f_1(x), f_2(x), ..., f_n(x)$ are “n” objective functions.

Concept of Domination: MOGA uses a concept of domination when comparing two solutions. If a feasible solution is not dominated by any other feasible solutions to a MOOP; the solution is said to be a non-dominated solution.

Domination: $x^{(1)}$ dominates $x^{(2)}$ if (1) $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives. (2) $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective. Solution $x^{(1)}$ is said to determine $x^{(2)}$ or $x^{(1)}$ is said to be non-dominated by $x^{(2)}$ if both above conditions are true [8].

Pareto–based ranking selection: The fitness values of the separate objectives are treated independently. Solutions are ranked into a “non-dominated” order in which the fittest ones are the solutions dominating the others that are less fit. The rank of each individual is calculated based on modified individual fitness. The ranking method is used such that if individual “$i$” dominates “$j$” other, then the rank of individual “$i$” is $1+n$.

B. Gender-Based Genetic Algorithms

General MOGAs are a nonsexual method that has the same sex of chromosomes. When reproducing, they do not respect the sex of chromosomes. This is called a nonsexual reproduction. On the other hand, sexual reproduction can be like humans that have two sexes: male and female. When creating the offspring, parents have to be chosen from both male individuals and female individuals to be the father and mother.

There are many papers that propose sexual reproduction in GA. Rejeb and Abuelhaija [9] propose a method that adds the special feature to chromosomes. The special feature is the sex or gender tag by representing “1” for male and “0” for female. The sexes of offspring are inherited from the parents that are both male and female. The mutation operator preserves the gender tag of chromosomes. Goh, Lim, and Rodrigues [10] describe a method that determines the sex of chromosomes randomly according to the problem. Every female individual has to be chosen once for creating the offspring. It uses tournament selection for selecting male individuals. Vrajitoru [11] proposes the different gender tags from others. He divided the sexes of chromosomes into four classes: male, female, self-fertilizing, and hermaphrodite. The concept of Sanchez-Velazco and Bullinaria [12] is similar to evolutionary of humans. That is they include both male and female as individuals in phynotypes. Their approach allows preferences to select the mates: direct fitness and in-direct fitness. For the mutation, male and female individuals have different mutation rates. The mutation rate of a male individual is higher than the female individual. Lis and Eiben [13] propose the algorithm that uses integer number to represent the sex of chromosomes. The numbers of sexes have to be equal to the number of criteria.

III. Multi-Objective Bisexual Reproduction Genetic Algorithm (MOBPGA)

Sexual reproduction has more varied genetic pool than a nonsexual reproduction because the genders preserve the diversity of genes and maintain a successful genetic pool by means of crossover, mutation and selection. This results in a new GA called a Multi-Objective Bisexual Reproduction Genetic Algorithm (MOBPGA). MOBPGA uses the concept of Darwin’s theory of sexual selection [10]. In general, the reproductive success of females is limited by resource availability. Male reproductive success is limited by the number of unfertilized females. If sexual selection works primarily on males, we would expect to see more variation in male mating success than in female mating success.

The schematic of the MOBPGA procedure, shown in Fig. 2, is similar to the concept of ($\mu+\gamma$) evolution strategy and NSGA-II [14]. NSGA-II uses crowding distance sorting to reject the lower fitness individuals. On the contrary, MOBPGA uses the concept of temporary non-dominated solutions, which is similar to SPEA [15]. SPEA keeps the non-dominated solutions in each generation external to the population. The population of solutions consists of male and female individuals. The numbers of male and female individuals are equal. The temporary solutions are non-dominated solutions that are...
found in every generation of population. Solutions are the Pareto optimal solutions that are selected from the temporary solutions. For the population, the individuals in the population are chosen to be the new population by using the roulette wheel selection. In the population, the numbers of male and female individuals may not be equal.

![Fig. 2 A Schematic of MOBRGA procedure](image)

In crossover operation, the parents are chosen from the new population by using the sexual selection. Crossover probability is used to measure which ones should be the parents. The parents consist of male and female individuals. If there is only one sex of individuals then there are no offspring. When creating offspring, the parents are chosen randomly. This is based on the crossover probability. Positions for crossover are also randomly chosen.

In mutation operation, the mutated individuals are chosen from the new population using different mutation rates for male and female individuals.

The population of the next generation consists of new offspring which are individuals resulted from crossover and mutation operations in the previous generation. The individuals are sorted according to objective function values or fitness values and rejected if individuals have low values. The number of population of next generation remains equal to \( N \). Fig. 3 illustrates the schematic of MOBRGA procedure.

![Fig. 3 Gene mutation](image)

In fact, there are two types of mutation operators. They are gene mutation and chromosome mutation. The gene mutation is the changing of genes and the chromosome mutation is the changing of structure or numbers of chromosomes. For MOBRGA, the mutated individuals are selected from the new population based on the sex of individuals and the mutation rate. For male individuals, they are mutated using a male mutation rate.

![Fig. 4 Chromosome mutation](image)

![Fig. 5 Flow chart of MOBRGA](image)
and the gene mutation that preserves the redundancy bit. The male mutation rate is 0.1. On the other hand, female individuals are mutated using a female mutation rate and the chromosome mutation. The female mutation rate is 0.01 that less than the male mutation rate. This means that the male individuals have more chance to mutate. For this stage, if the parents contains only one sex of individuals then there is no affect to the mutation operation because the mutated individuals may be either both sexes or only one sex. Fig 3 and 4 show the different mutation types.

Fig. 5 shows the flow chart of MOBRGA. The procedures are described below.

**Step 1**: generate the initial population: male and female individuals.

**Step 2**: evaluate each individual by decoding chromosomes to the problem domain, ranking individuals, and calculating fitness.

**Step 3**: sort the temporary non-dominated solutions. Roulette wheel selection is used to select the new population. The parents are selected using sexual-based selection.

**Step 4**: create the offspring by using crossover and mutation operators.

**Step 5**: determine the termination criteria. If the criteria are met, then sort the Pareto optimal solutions from the temporary solutions.

**IV. Experimental Results**

To prove the concept of MOBRGA, simulation studies were performed. For comparisons, MOBRGA was compared with other three algorithms: Non-dominated Sorting Genetic Algorithm (NSGA-II) [14], Strength Pareto Evolutionary Algorithm (SPEA2) [15], and Multi-Objective Extension of a Popular Particle Swarm (MOPSO) [16].

For multi-objective optimization problems, we would like to compare MOBRGA with many kinds of EA that is not focused on only nonsexual GAs or sexual GAs. For comparison, we used three well-known benchmark problems; Shaffer’s, Murata’s, and Srinivas and Deb’s problems. The first two problems were used for comparing with KEA. The latter was used for comparing with a Multi-Sexual Genetic Algorithm (MSGA) [13] as it is a multi-objective optimization problem. Murata’s problem is a function with a contiguous Pareto set and its Pareto front is also contiguous but not convex. Shaffer’s problem is a function with a contiguous Pareto set and its Pareto front is also contiguous and convex. For Srinivas and Deb’s problem, it is a complicated function where the result of a Pareto front is convex but it is different from the concept of a Pareto front.

**Benchmark Problems**

- **Shaffer’s problem**
  
  Minimize: 
  
  \[ f_1 = x^2 \] 
  \[ f_2 = (x-2)^2 \] 

  Constraint: \( 0 \leq x \leq 20 \)

- **Murata’s problem**
  
  Minimize: 
  
  \[ f_1 = 2\sqrt{x_1} \] 
  \[ f_2 = x_1(1-x_2) + 5 \] 

  Constraint: \( 1 \leq x_1 \leq 4 \); \( 1 \leq x_2 \leq 2 \)

- **Srinivas and Deb’s problem**
  
  Minimize: 
  
  \[ f_1(x,y) = (x^2+y^2)^{1/8} \] 
  \[ f_2(x,y) = ((x-0.5)^2+(y-0.5)^2)^{1/4} \] 

  Constraint: \(-5 \leq x \leq 10\); \(-5 \leq y \leq 10\)

To find solutions to the problems the parameters were set as follows: Population size = 80, Generation = 100, Crossover rate = 0.7, Mutation rate for male individuals = 0.1, and Mutation rate for female individuals = 0.01.

For comparisons, Shaffer’s problem is used to solve the Pareto front solution using MOBRGA, MOPSO, NSGA-II, and SPEA2. Shaffer’s problem is a two-function minimization problem. It contains only one constraint variable from interval 0 to 20.

Murata’s problem contains two variables. It consists of two functions for minimization, similar to Shaffer’s problem. The constraint values of variables are different. The first variable is from 1 to 4; the second variable is from 1 to 2. It is used for comparing MOBRGA, MOPSO, and NSGA2.

Srinivas and Deb’s problem, a well-known reference in gendered GA, is a complicated function. It contains two variables that it has the constraint value only one variable from -5 to 10. It is used for sexual selection comparing MOBRGA and MSGA.

**Shaffer’s problem testing results**

The results from all algorithms could cover the region of solutions and efficiently display the Pareto front. MOBRGA could propagate the Pareto optimal solutions very nicely; however, in some parts, it could not show non-dominated solutions. MOBRGA was better at displaying the Pareto front than NSGA-II, SPEA2, and MOPSO, because it could find non-dominated solutions where other algorithms could not. On the contrary, it had more replicated solutions that were showed on the Pareto front. Fig. 6 shows the Pareto front of Shaffer’s problem using MOBRGA.

**Murata’s problem testing results**

NSGA-II could cover a wider region of solutions than others. It could show Pareto optimal solutions almost continuously and it had few replicated solutions. Fig. 7 illustrates the Murata’s problem solved using MOBRGA. MOBRGA could show the Pareto front but it was not
continuous. It could cover the region of solutions, like NSGA-II, but solutions were congested in only some regions of the Pareto front. MOBRGA had more replicated solutions than others, but it was better than MOPSO, which had wider space in some Pareto front regions.

Srinivas and Deb’s problem testing results

From the comparison, MSGA could show the Pareto front better than MOBRGA because the result of MSGA could cover the solutions region wider than MOBRGA and they were continuous. The solutions of MOBRGA were not continuous and they were congested and replicated in some regions of the Pareto front. Fig. 8 illustrates the Pareto front of Srinivas and Deb’s problem using MOBRGA.

MOBRGA was applied to the design of a computer network. Small and medium business design and Cisco products were considered. The solutions to computer network designs have to support users’ requirements. Decision makers can choose an optimal solution from the set of solution resulting from MOBRGA.

Genotype for Computer Network Design: The chromosome format is shown in Fig. 9. It consists of types of routers, types of firewalls, types of switches, the numbers of switches, and the sex bit.

TR, denotes the types of routers \((l = 1, \ldots, L)\).

TF, denotes the types of firewalls \((i = 1, \ldots, M)\).

TS, denotes the types of switches \((j = 1, \ldots, N)\).

NS, denotes the number of switches \((k = 1, \ldots, O)\).

RB denotes the redundancy bit or sex bit \((RB \in \{0,1\})\).

Multi-Objective Functions: There are many cases of solutions that are suitable for different requirements. The objective functions of MOBRGA are minimizing cost and maximizing supported users. We also have to choose the cases of solutions according to user’s requirements. The algorithm has to the minimize the total cost: Cost = (Prices of Routers) + (Prices of Firewall) + (Prices of Switch × Numbers of switches). In addition, the algorithm has to maximize the numbers of users: Numbers of users = Total ports of switches.

Computer Network Design Result

Fig. 10 shows the result of a designed computer network in the searching space. It shows that there are seven solutions which are indicated as points. Each point or solution consists of a specification of switches, the number of switches, total price, and total users. They are non-dominated solutions. These switches support user application requiring a file and printer sharing. They may be different in types, prices, ports, and numbers of switches. For some solutions, they are the same types but different in the numbers of switches.
Fig. 10 Result of computer network design

These solutions in Fig. 10 are according to objective functions that are minimization of cost and maximization of users. For example, the first solution (the first point in the diagram) is WS-C2940-8TF, one of CISCO products. It can support voice services and it can connect to a 10/100 Ethernet LAN. There is a switch that can support eight users. Its total price is 39,800 baht (995 US$).

VI. Conclusion

This paper describes a novel multi-object bisexual reproduction genetic algorithm (MOBRGA). The proposed algorithm uses Darwin’s concept which focuses on male individuals and it uses the crossover operator according to the number of male individuals, which have higher mutation rate than female. It means that male individuals get more chance to mutate.

The experiments confirm the ability of MOBRGA for multi-objective optimization problems to find a set of Pareto optimal solutions. MOBRGA could find Pareto optimal solutions in some problems such as Shafer’s problem and Murata’s problem. It uses sexual selection and different mutation rates and types for creating the new population. It is like humans that create offspring from a father and a mother that are of different sex.

For a computer network design problem, MOBRGA could find non-dominated solutions. It was based on the concept of a small and medium business of the Cisco Company. There were four specific requirements: routers, firewalls, switches, and applications. Users had to specify the requirements and then MOBRGA used them to find the design solutions. The solutions were depended on the numbers of binary solutions coding and the conditions of solutions. There were many solutions if many bits of binary representation were used.

An advantage of MOBRGA is non-limitation of criteria because it uses the same optimization criterion for the different sexes of individuals, and storing the set of non-dominated solutions that may exceed the population size. It can show the Pareto front that covers the feasible region in the objective space. On the other hand, it has a problem with premature convergence; sometimes it can not find the global solutions. This problem is difficult to solve because GA uses a random search technique and random initial population. It takes a long time to find Pareto optimal solutions.

Future Research

There are several ways to continue this path of research. MOBRGA can use different fitness functions between male and female individuals because male and female individuals are different characteristics. As the result, the solutions may converge faster. The sex of offspring can be inherited from the parents according to the fitness of parents. In this way, it may speed the convergence time too. Usually GAs are very time consuming in searching for solutions. Grid computing can help GAs to reducing the searching time. In other words, a parallel genetic algorithm that processes by many processors can be applied to find the solutions simultaneously. In the network design, it can be applied to wired LAN, wireless LAN, and other complicated network such as WAN and MAN architectures. Finally, the proposed algorithm can be applied to other application such as in fuzzy system designs, neural network designs, and others.

References