A Multi-objective Genetic Algorithms Method For Generating Pareto Solutions in Bilateral Negotiations

Zichen Yang
School of Computer Science
Wuhan University
Wuhan, Hubei, 430072, P. R. China

Bo Meng
School of Computer Science
Wuhan University
Wuhan, Hubei, 430072, P. R. China

Abstract – Genetic algorithms are often well suited for multi-objective optimization problems. This paper reviews the methods for generating Pareto optimal solutions in bilateral negotiations. A method is put forward for generating Pareto solutions using multi-objective genetic algorithms (MOGA); software is developed with Visual C++. Through a simulation study the method is proven to be practical and effective.

I. INTRIDUCTION

Conflicts among decision makers can arise from differences in interests or in objectives, and from cognitive limitations. The need for negotiations arises from two different types of disputes: (1) Negotiators' interests are fundamentally opposed, (2) Negotiators share basic objectives but they differ in their assessment of the priorities of the objectives. Negotiation is one of the best approaches for conflict resolution. Negotiation is a process in which two or more parties with conflicting objectives attempt to reach an agreement. Sebenius [1] first put forward concept of negotiation analysis. Many researchers have been make studies based on Sebenius's idea. With the Internet dramatically development, Internet-based negotiations are becoming one of the forms of business communication. The traditional concept of negotiation is extended by web-based negotiation support systems (WNSS). WNSS are designed to assist negotiators in reaching mutually satisfactory decisions by providing a means of communication and through analysis of available information.

In the current literature, the methods for generating Pareto solutions in bilateral negotiations can be divide two categories: (1) The method of improving directions [2], there are three steps in the method of improving direction: First: Identify parties improving directions (e.g., in terms of their most preferred directions) at the tentative agreement. Second: Choose a compromise direction at that point. Third: Choose a new tentative agreement along the compromise direction. The process would continue until the Pareto solutions are searched. (2) The method of constraint proposal [3], the mediator chooses a reference point and arbitrary plane going through it. He declares the plane constraint to the parties who give their optimal alternatives on the plane. Each of party resolves optimization problem by himself. If the distance between the optimal alternatives is small enough or the optimal alternatives coincide, then the solutions are considered Pareto solutions. Otherwise, continue. These methods had been proved in the literature [2][3][4].

In this paper, we will present the genetic algorithms-based multi-objective for generating Pareto solutions in bilateral negotiations. The purpose of this paper attempts a new way so that the method can be adopted by negotiation analysis. This paper is organized as follows: Section II introduces the basis concept of Pareto solutions and describes the details of multi-objective genetic algorithms (MOGA). Section III gives a simulation study. Section IV contains a conclusion.

II. MOGA

From a more general decision-making standpoint, negotiation process can be considered as decision-making process. In order to gain Pareto agreement or satisfying agreement, several objects needed to be considered. Combining these objectives form a multi-objective optimization problem. Pareto-efficiency and joint feasibility are key concepts in the economic approach to analyzing negotiations. Pareto optimality is a measure of

0-7803-7268-9/01/$10.00

negotiation efficiency [5]. Genetic algorithms have been proved considerable successful in providing Pareto solutions to many complex multi-objective optimization problems and get more and more attentions.

Genetic algorithms (GAs), first formalized as an optimization method by Holland, are search algorithms based on the mechanics of natural selection. The GAs differ from most optimization techniques in four ways: (i) GAs search with a population of points, not a single point. (1) GAs use only the value of the function information, and not derivatives or other auxiliary knowledge. (2) GAs work with a coding of a parameter set not the parameters themselves. (3) GAs use randomized parents selection and crossover from the old generation. The combination of multi-objective programming and genetic algorithms to explore multi-objective genetic algorithms has significant meaning in bilateral negotiations for generating Pareto solutions.

A. Definition of Pareto optimal

In this section, we describe a multi-objective optimization problem and the concept of Pareto optimal solution. Eq. (1) represents the multi-objective problem mathematically.

$$\begin{aligned} &Minimize/Maximize: \quad F(x)=[f_1(x), f_2(x), \cdots, f_P(x)] \\ &Subject\ \ to \quad g_i(x) \le 0, i=1,2,\cdots m. \end{aligned} \tag{1}$$

where $F(x)$ is multi-objective function vector, $f_l(x)$ is lth objective function, x is input vector, P is the number of objective functions, m is the dimension of input vector. A solution $x^* \in X$ is said to be Pareto optimal or an efficient solution or non-dominated or a non-inferior point for VOP if and only if there is no $x \in X$ such that $f_l(x) \le f_l(x^*)$ for all $l \in \{1,2,\cdots,P\}$, with at least one strict inequality.

B. Fitness function

In this paper, we adopt weighted sum method, which transforms multi-objective optimization problem into a scalar optimization problem as Eq. (2). We assign random weighting coefficients for each objective function.

$$F(x)=w_1 f_1(x)+w_2 f_2(x)+\cdots+w_P f_P(x)=\sum_{i=1}^{P} w_i f_i(x) \tag{2}$$

where $F(x)$ is fitness of genetic algorithms, $f_i(x)$ is ith objective function, P is the number of objective functions, w_i is weighting coefficient, w_i is a non-negative random number which satisfy the following relations (2) and (3).

$$w_i = random(x_i) \Big/ \sum_{i=1}^{P} random(x_i) \tag{3}$$

$$w_i \in [0,1] \quad \sum_{i=1}^{P} w_i = 1 \tag{4}$$

The fitness function is utilized when a pair of parent solution is selected for generating a new solution by crossover and mutation. The basic algorithm proposed by the author is the following (see Fig. 1). More details are described in the following sections.

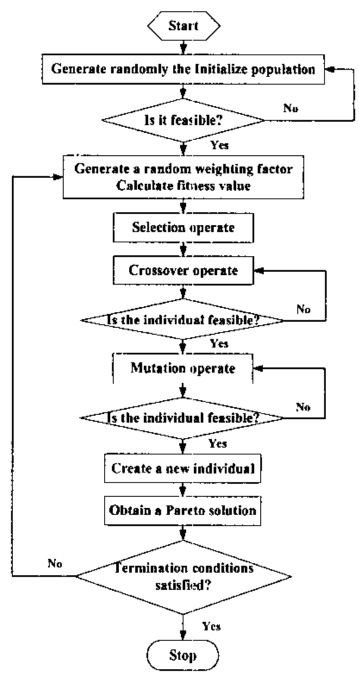

Fig. 1. Flow chart of the main

C. Encoding scheme

We can apply a binary vector as a chromosome to represent real value of decision variable, but the required precision is depended on length of the vector. On the other hand, when we use a binary vector as chromosome to represent real value of decision variable in multi-dimensions or high- precision problem, search space will be hugeness. An alternative approach to represent a solution is the floating-point implement in which each chromosome vector is coded as a vector of floating number avoiding encoding and decoding, accelerating search.

D. Initial population

The initial population is fundamental work for genetic algorithms. In order to ensure feasibility and diversity of the individual, we first determine an interior point, denoted by X_0 in the constraint set S and define a large positive number M_0. We can succeed initial population according to following step [6]:

(1) Randomly select a direction d in R^n,

(2) Obtain an individual X if $X = X_0 + M_0 * d$ is feasible; Otherwise, set M_0 as a random real number between 0 and M_0 until is feasible.

Repeat the above process $popsize$ times, we can obtain $popsize$ initial feasible solutions.

E. Evaluation function and selection process

Evaluation function is an important aspect in determining the success or failure of a GA. In this paper, we employ the well-known rank-based evaluation function [8]:

$$Eval_{rank} = \alpha * (1-\alpha)^{rank-1}$$
$$rank = 1,2,\cdots, popsize \tag{5}$$

where $rank$ is the ordinal number of the lth individual X in the rearranged list, $rank = 1$ means that the best individual, and $rank = popsize$ means the worst one, α means selection pressure, the bigger selection pressure, the higher probability is selected. The individual with higher selection pressure will have more opportunity to produce offspring.

Selection process is one of the most fundamental genetic operators. The selection process is based upon the roulette wheel $popsize$ times, and each time select an individual for a new population according to following step:

(1) Calculate the cumulative probability q_i,

$$q_i = \sum_{i=1}^{popsize} Eval(X_i) \tag{6}$$

(2) Generate a random real number r, $r \in [0,1]$,

(3) If $r < q_1$, the first individual will be selected, otherwise, if $q_{i-1} < r < q_i$, the ith individual will be selected, Repeat (2) and (3) $popsize$ times accomplish selection process.

F. Crossover

Crossover operation is widely considered as critical to the success of GA. Crossover operates on two parents at a time and generates offspring by recomposing both parent features. This means that two individual of the population exchange genes. There are many ways of implementing crossover operation, for example having a single crossover point. In this paper, we adopt arithmetical crossover operation to produce two children x_1' and x_2' as follow [6]:

$$\begin{cases} x_1' = \lambda_1 x_1 + \lambda_2 x_2 \\ x_2' = \lambda_1 x_2 + \lambda_2 x_1 \end{cases} \tag{7}$$

where x_1' and x_2' are children, x_1 and x_2 are parents, λ_1, λ_2 are random real number, and also $\lambda_1, \lambda_2 \geq 0$, $\lambda_1 + \lambda_2 = 1$. We must check the children value so that they are feasible. Implement process of crossover operation is similar to SGA's crossover operation [6][8].

G. Mutation

Mutation refers to the process of increasing diversity in the population by introducing random variations in the members of the population. We adopt the method similarly initial population process as follow [6][8]:

(1) Randomly select a direction d in R^n,

(2) Obtain child X_m' if $X_m' = X_m + M_0 * d$ is feasible, otherwise, set M_0 as a random real number 0 between M_0 until $X_m + M_0 * d$ is feasible.

where X_m is parent, denoted by $[x_1, x_2, \cdots, x_n]$, X_m' is child.

We must check the children value so that they are feasible. Implement process of mutation operation is similar to SGA's mutation operation.

H. Elitist strategy

The idea of elitist strategy is that the maximum fitness value of individual don't take part in operating of crossover and mutation in current population, but utilize it to replace the minimum fitness value of individual by crossover and mutation operation in this population [6][8]. The steps describe as follow:

(1) Find out the maximum and minimum fitness value of individual in current population,

(2) If the best individual in current population is higher than the best individual up to now, thus it will be the best individual,

(3) Utilize the best individual to replace the worst individual up to now.

In this paper, we set five best individual in the every iteration.

III. SIMULATION

The goal of simulation is that GAs can be utilized to generate Pareto solutions in bilateral negotiations. The demonstration is cited from literature [7] (**Case II**). The problem: two competitors X and Y face the option of assigning a certain fraction of their investments to two facilities A and B. The costs to the two competitors are determined by how much the two assign to the two facilities. The cost functions for the two competitors are described as follow:

$$\min \quad C_x = pe^{\ln(3)pq} + 2(1-p)e^{\ln(2)(1-p)(1-q)}$$

$$\min \quad C_y = qe^{\ln(3)qp} + 2(1-q)e^{\ln(2)(1-q)(1-p)}$$

$$0 \le p \le 1 \quad 0 \le q \le 1 \tag{8}$$

The fitness function $F(x)$ is specified as $F(x) = -w_1 C_x - w_2 C_y$. Parameters selected for this simulation are as follows: the population size $popsize = 100$, the number of iteration $MaxGens = 100$, the probability of crossover $p_c = 0.8$, the probability of mutation $p_m = 0.01$, the number of selection pressure $\alpha = 0.1$, the number of elite individual $N_{elite} = 5$, an interior point $X_0 = (0.5, 0.5)$, the large positive number $M_0 = 50$. The final solutions are clearly showed in Fig. 2. From the simulation result, we not only obtain results same as **Case II**, but also gain more Pareto solutions for negotiations.

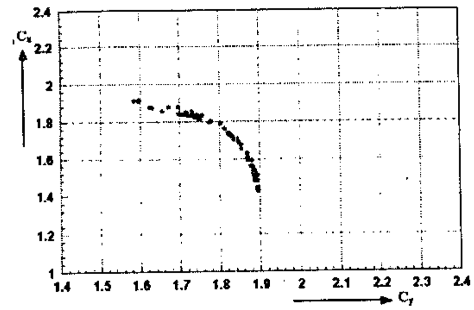

Fig. 2. Pareto solutions

IV. Conclusion

In this paper we introduced the relationship between negotiation problem and multi-objective optimization, and successfully applied MOGA to generate Pareto solutions in bilateral negotiations. A simulation study is tested to prove the method is practical and effective. Future work will address how to elicit preference of negotiator and analyze data and support expression of the preference information by visual mode.

V. ACKNOWLEDGMENTS

This research is sponsored by National Ministry of Education of China and Natural Science Foundation of Hubei province (2001ABB058).

VI. REFERENCES

[1] James K.SEBENIUS, "Negotiation Analysis: A Characterization And Review", Management Science, Vol. 38, No. 1, January 1992.

[2] Harri Ehtamo, Raimo P. Hamalainen, "Generating Pareto Solutions in a Two-party Setting: Constraint Proposal Methods", Management Science, Vol. 45, No. 12, December 1999.

[3] Harri Ehtamo, Eero Kettunen, Raimo P. Hamalainen, "Searching for joint gains in multi-party negotiations", European Journal of Operational Research 130 (2001) 54~69.

[4] H.Ehtamo, M.Verkama, and R.P.hamalainen, "On distributed computation of Pareto solutions for two decision makers", IEEE Trans. Systems, Man, and

Cybernetics Part A: Systems and Humans, Vol. 26, No. 4, pp.498~503, 1996.

[5] Barry Blecherman, "Adopting automated negotiation", Technology In Society 21 (1999) 167-174.

[6] Mitsuo Gen and Runwei Cheng, Genetic Algorithms and Engineering Design, Science Publishing, 2000.

[7] GIAMIERO E.G.BEROGGI, "Negotiation and Equilibria in User Competition for Resources: A Dynamic Plot Approach", Computational & Mathematical Organization Theory 6, 61-82, 2000.

[8] Zbigniew Michalewicz, Evolution Programs, Science Publishing, 2000.