Game Model Based Co-evolutionary Algorithm and Its Application for Multiobjective Nutrition Decision Making Optimization Problems

Gaoping Wang and Liyuan Bai

School of Information Science and Engineering , Henan University of Technology,
450052 Zhengzhou, China
wgp888888@163.com

Abstract. Sefrioui introduced the Nash Genetic Algorithm in 1998.This approach combines genetic algorithms with Nash's idea. Another central achievement of Game Theory is the introduction of an Evolutionary Stable Strategy, introduced by Maynard Smith in 1982. In this paper, we will try to find ESS as a solution of MOPs using our game model based co-evolutionary algorithm.We present A Game model based co-evolutionary algorithm (GMBCA) to solve this class of problems and its performance is analyzed in comparing its results with those obtained with four others algorithms. Finally, the GMBCA is applied to solve the nutrition decision making problem to map the Pareto-optimum front. The results in the problem show its effectiveness.

1 Introduction

In Multi-objective optimization problems (MOPs), the aim is to simultaneously optimize a group of conflicting objectives. MOPs are a very important research topic, not only because of the multi-objective nature of most real-world decision problems, but also because there are still many open questions in this area. The traditional optimization problems attempt to simultaneously minimize cost and maximize fiscal return. However, in these and most other cases, it is unlikely that each objective would be optimized by the same parameter choices. Hence, some trade-off between the criteria is needed to ensure a satisfactory design. In searching for solutions to these problems, we find that there is no single optimal solution but rather a set of solutions. These solutions are optimal in the sense that no other solutions in the search space are superior to them when all objectives are considered.They are generally known as Pareto-optimal solutions[1].

In this paper, we present the GMBCA [2] and we analyze it regarding the solution of MOPs. Moreover, we compare its results with those obtained by the multiobjective evolutionary algorithms (MOEAs): VEGA [3], NPGA [4] and MOGA [5], and the classical method of objective weighting refereed as [1]. We compare its performances in the solution of two analytical test problems. Finally, we apply the GMBCA to solve the nutrition decision making problem with the aim of find the optimum tradeoff surface.

Y. Wang, Y. Cheung, and H. Liu (Eds.): CIS 2006, LNAI 4456, pp. 177–183, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Multiobjective Optimization Problem

Consider a MOP model as presented below:

$$\begin{aligned} &\text{Optimize} && y=f(x)=\{f_1(x),f_2(x),f_3(x),\ldots,f_m(x)\} \\ &\text{subject to} && G(x)=\{g_1(x),g_2(x),\ldots,g_j(x)\}\leq 0 \\ & && h(x)=\{h_1(x),h_2(x),\ldots,h_k(x)\}=0 \\ & && \text{where} \quad x=\{x_1,x_2,\ldots,x_N\}\in X \\ & && y=\{y_1,y_2,\ldots,y_m\}\in Y. \end{aligned} \tag{1}$$

And x is the vector of decision variables, y is the objective vector, X is the decision space, and Y is called the objective space. vectors G(x) and H(x) represent problem's constraints.In MOPs, the aim is to find the optimal solution x' $\in$ X which optimize f(x). Each objective function, fi(x), is either maximization or minimization.

3 A Game Model Based Co-evolutionary Algorithm (GMBCA)

3.1 Nash Genetic Algorithm (Nash GA)

The idea of Nash GA is to bring together genetic algorithms and Nash strategy in order to cause the genetic algorithm to build the Nash Equilibrium. In the following, we present how such merging can be achieved with 2 players trying to optimize 2 different objectives.

Let s = XY be the string representing the potential solution for a dual objective optimization problem.Then X denotes the subset of variables handled by Player 1 and optimized along criterion 1. Similarly Y denotes the subset of variables handled by Player 2 and optimized along criterion 2. Thus, as advocated by Nash theory, Player 1 optimizes s with respect to the first criterion by modifying X while Y is fixed by Player 2. Symmetrically, Player 2 optimizes s with respect to the second criterion by modifying Y ,while X is fixed by Player 1.

The next step consists of creating two different populations, one for each player. Player 1's optimization task is performed by Population 1 whereas Player 2's optimization task is performed by Population 2. Let X_{k-1} be the best value found by Player 1 at generation k –1 and Y_{k-1} be the best value found by Player 2 at generation k –1 . At generation k , Player 1 optimizes X_k while using Y_{k-1} in order to evaluate s (in this case,.s = $X_k Y_{k-1}$). Simultaneously, Player 2 optimizes Y k while using X k- 1 in order to evaluate s (in this case, k= $X_{k-1} Y_k$). After the optimization process, Player 1 sends the best value X k to Player 2 who will use it at generation k +1 . Similarly, Player 2 sends the best value Y_k to Player 1 who will use it at generation k+1 . Nash equilibrium is reached when neither Player 1 nor Player 2 can further improve their criteria [6].

3.2 Evolutionary Stable Strategy (ESS)

The primary contribution of evolutionary game theory (EGT) is the concept of the Evolutionary Stable Strategy (ESS). ESS was originally proposed by a world renowned biologist named Maynard Smith based on EGT and defined as an unchangeable strategy by other strategies. Unchangeable strategy means that no matter how outstanding a particular strategy may be, it cannot maintain predominance over other inferior strategies permanently. In the context of an actual ecosystem,more evolutionary stable species can be reserved than superior species, in other words an evolution chooses the strategy that not only executes progressive direction but also moves the equilibrium state.

3.3 A Game Model Based Co-evolutionary Algorithm (GMBCA)

In this section, the co-evolutionary algorithm designed for searching ESS of MOP is explained.Throughout the game, players for each objective function try to optimize their own objectives and all individuals in a population set are rewarded. The reward value is determined by the percentage of victories during the game .

To design the co-evolutionary algorithm based on Game Theory (GMBCA), we first established a game player with randomly generated populations. All individuals in each population are rewarded 'fitness' that will be used during the selection procedure. During the game each individual in the first population plays the game with others in the remaining populations and is paid the fitness . Other individuals in the remaining populations execute the game in the same manner by turns. Using the fitness,the next generation individuals are produced in each population independently through crossover and mutation.

Step 1: Two populations are randomly generated .

Step 2: The first individual in the primary population plays with each individual in the other population and is evaluated for level of fitness.

Throughout the game by turns, the fitness of the opponent individual in the second population is calculated in the same manner.

Step 3: The process of Step 2 is executed for all individuals of the first population one by one.

Step 4: The processes of Step 2 and Step 3 are executed for all individuals of the second population analogously.

Step 5: Using Fitness(xi) and Fitness(yj) determined from the previous procedures, each population produces next generation individuals independently through crossover and mutation.

Step 6: Until ending condition is satisfied the procedures from Step 2 to Step 5 are reiterated.

4 Description of VEGA, MOGA, and NPGA

An early GA application on multiobjective optimization by Schaffer opened a new avenue of research in this field. The algorithm,called vector evaluated genetic algorithm (VEGA), performs the selection operation based on the objective switching rule, i.e., selection is done for each objective separately, filling equally portions of mating pool [3]. Afterwards, the matting pool is shuffled, and crossover and mutation are performed as usual.

Fonseca and Fleming [5] proposed a Pareto-based ranking procedure (MOGA), where the rank of an individual is equal the number of solutions found in the population where its corresponding decision vector is dominated. The fitness assignment is determined by interpolating the fitness value of the best individual (nondominated) and the worst one (most dominated).The MOGA algorithm also uses a niche-formation method to distribute the population over the Pareto-optimal region based on the objective space.

The niched Pareto genetic algorithm (NPGA) proposed by Horn, Nafpliotis, and Goldberg uses the concept of Pareto dominance and tournament selection in solving MOPs [4].In this method, a comparison set of individuals is randomly picked from the current population before the selection procedure. In addition, we choose two candidates from the current population that will compete to survive to the selection operation. For selecting the winner, these two candidates are compared with those of set using a nondomination criterion as described in Section 2.

5 Criterion for Performance Measurements

The performance measurement criterion [7,8] used to evaluate the Pareto fronts produced by the EAs is the coverage relationship. Given two sets of nondominated solutions, we compute for each set the fraction of the solutions that is not covered (not dominated) by the other. Since this comparison focus on finding the Pareto-optimal set, this criterion uses the off-line performance method. The nondominated solution set taken to perform the comparison between all EAs is the summation of nondominated solutions found by each algorithm at each run, after application of a nondominance criterion.

6 Criterion for Performance Measurements

6.1 Test Problems

The algorithm is tested on the following problem. The problem was collected from Deb :

Two problems[9] were chosen in order to test the multiobjective genetic algorithms discussed in this paper. The problem1 has a convex Pareto-optimal front and is given by

$$F_1(x_1, x_2, \ldots, x_m) = x1$$
$$F_2(x_1, x_2, \ldots, x_m) = g(x) * (1 - \sqrt{F1/G(x)}) . \qquad (2)$$

The second problem is the nonconvex counterpart to problem1

$$F_1(x_1, x_2, \ldots, x_m) = x1$$
$$F_2(x_1, x_2, \ldots, x_m) = G(x) * (1 - (F1/G(x)^2)) . \qquad (3)$$

In both cases, m=30,xi∈[0,1], and the Pareto-optimal front is formed with G(x)=1. The function G(x) is defined by

$$G(x_1, x_2, \cdots, x_m) = 1 + 9\sum_{i=2}^{m} x_i /(m-1) \qquad (4)$$

6.2 Experimental Results and Discussions

The multiobjective EAs were executed 30 times for each problem with the same initial population. The results of each execution was stored in an auxiliary vector and at the end the nondominance criterion was applied to the points belonging to the auxiliary vector, resulting a nondominated set that was taken as outcome. The set of genetic parameters used are: N_{ger}=250, N_{pop}=100, P_c=0.8, P_m=0.01, Ashared=0.4886 , T_{dom}=10 and (for NPGA). The graphic results are shown in Figs. 2 and 3.

The direct comparison of the outcomes achieved by the different multiobjective EA is presented in Table I. Each cell gives the percentage of solutions evolved by method B that are nondominated by those achieved by method A for both problems and . For example, the cell NPGA/MOGA signifies that 90% of solutions found by NPGA are nondominated by those found by MOGA for problem and 89% in the case of problem.These results show that all methods give rise to similar solutions with a slight superiority for GMBCA method, with exception of VEGA. The result for VEGA method is explained by the fact of its selection procedure does not use information of nondominated fronts.

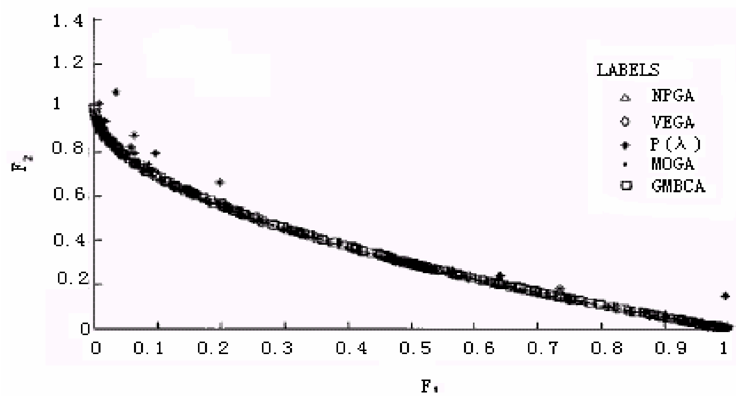

Fig. 1. Nondominated points for problem 1

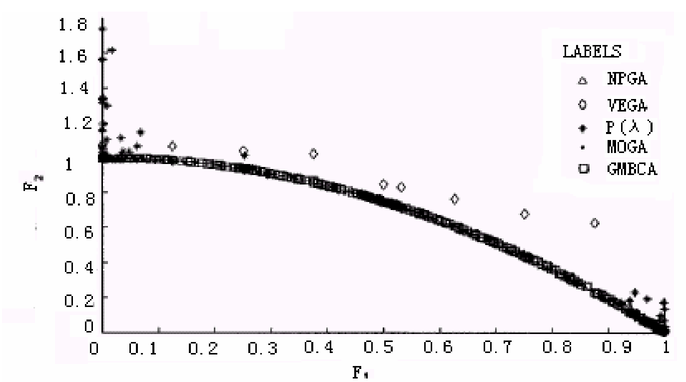

Fig. 2. Nondominated points for problem 2

6.3 Optimization in Nutrition Decision Makinings

The nutrition decision making Problem was chosen to show the application of GMBCA described in the previous study in solving a multiobjective nutrition optimization problem. In this paper, we search to find the Pareto-optimal front of a nutrition decision making .The aim is to find the multiple Pareto-optimal points considering two objective functions: 1) the first objective functions considers the energy and 2) the second one takes into account the protein. The constraint conditions are the bounds in the design variables.

Mathematically, the multiobjective optimization problem for the nutrition decision making problem was stated as

$$F=\min\{F_1,F_2\}=\min\{\text{abs}((e(x)-e0)/e0), \text{abs}((p(x)-p0)/p0)\} \quad (5)$$

The problem was solved considering seven design variables in continuous case . The nondominated points have been found using GMBCA method (with roulette wheel selection,and) coupled with a finite element code for energy and protein calculations. The domain was subdivided in these elements of first order.The results are presented in Figs.3 .

Table 1. EA performance measurement

B/A	VEGA	MOGA	GMBCA	NPGA	P(λ)
VEGA	-	0/0	0/0	0/0	0/0
MOGA	100/100	-	90/89	100/98	100/100
GMBCA	100/100	100/100	-	100/100	100/100
NPGA	100/100	90/89	90/88	-	100/100
P(λ)	100/100	88/79	87/90	79/89	-

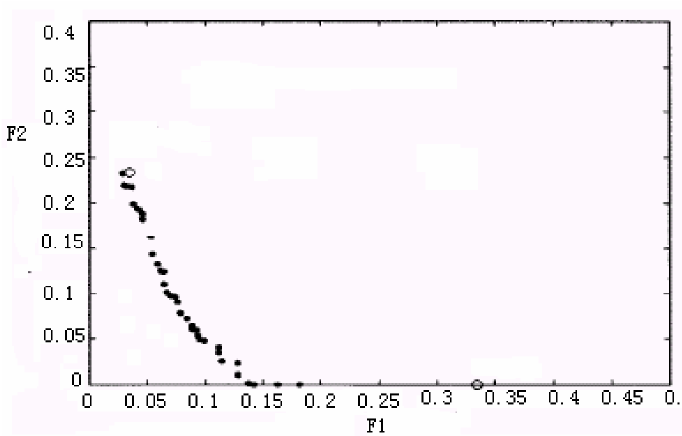

Fig. 3. Pareto-optimal points for the nutrition decision making problem

7 Conclusions and Future Research

In this paper, a co-evolutionary algorithm based approach is presented for multiobjective optimization problems. We tested the approach on two benchmark problems and it was found that our approach is promising when compared to a standard approach from the literature. Its application to the nutrition decision making problem show that it is reliable to solve multiobjective optimization in nutrition.For future work, we intend to test the algorithm on more problems.

References

1. Abbass, H., Sarker, C.R.: A pareto differential evolution approach to vector optimization problems. Congress on Evolutionary Computation 2, Newton, pp. 971–978 (2001)
2. Deb, K.: Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Evolution Computation 7, 205–230 (1999)
3. Coello, C.: A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information Systems 12, 269–308 (1999)
4. Deb, K., Goel, T.: Multi-objective evolutionary algorithms for engineering shape design. In: Sarker, R., Mohammadian, M., Yao, X. (eds.) Evolutionary Optimization, Kluwer Academic Publishers, USA (2001)
5. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. On Evolutionary Computation 3(4), 257–271 (1999)
6. Sefrioui, M., Periaux, J.: Nash genetic algorithms:examples and applications. In: Proc. Congress on Evolutionary Computation, pp. 509–516. IEEE Press, New York (2000)
7. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - a comparative case study. In: Proc. Fifth Int. Conf. On Parallel Problem Solving from Nature (PPSN-V), pp. 292–301. Springer, Berlin, Germany (1998)
8. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans.on Evolutionary Computation 3(4), 257–271 (1999)
9. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. In: Proc. 1999 Genetic and Evolutionary Computation Conf. Workshop Program, pp. 121–122. Orlando, Florida (1999)