Coevolutionary Optimization Algorithm: With Ecological Competition Model

Jianguo Liu and Weiping Wu

Congqing Business and Technology University ,
400067 Chongqing, China
ljg65@163.com, wuweiping@ctbu.edu.cn

Abstract. Premature convergence and low converging speed are the distinct weaknesses of the genetic algorithms. a new algorithm called ECCA (ecological competition coevolutionary algorithm) is proposed for multiobjective optimization problems, in which the competition is considered to be in important position. In the algorithms, each objective corresponds to a population. At each generation, these populations compete among themselves. An ecological population density competition equation is used for reference to describe the relation between multiple objectives and to direct the adjustment over the relation at individual and population levels. The proposed approach store the Pareto optimal point obtained along the evolutionary process into external set, enforcing a more uniform distribution of such vectors along the Pareto front. The experiment results show the high efficiency of the improved Genetic Algorithms based on this model in solving premature convergence and accelerating the convergence.

Keywords: Competitive coevolutionary genetic algorithm, Pareto optimal point, multiobjective optimization problems.

1 Introduction

This solution of a real problem involved in multiobjective optimization (MO) must satisfy all optimization objectives simultaneously, and in general the solution is a set of indeterminacy points. The task of MO is to estimate the distribution of this solution set, then to find the satisfying solutions in it. General MO contain a set of *n* decision variables, a set of *k* objective functions, and a set of m constraints. In this case, objective functions and constraints respectively become functions of the decision variables. If the goal of multiobjective optimization problems is to maximize the objective functions of the y vector, then (1):

$$\begin{aligned} &maximize\ \ \mathrm{y}=\mathrm{f(x)}=(\mathrm{f_1(x)},\cdots,\mathrm{f_i(x)},\cdots \mathrm{f_k(x)}), \\ &subject\ \ \ \mathrm{e(x)}=(\mathrm{e_1(x)},\cdots,\mathrm{f_j(x)},\cdots \mathrm{f_m(x)})\le 0 \end{aligned} \tag{1}$$

Where x=(x_1, x_2,…,x_n)$\in X$, y= (y_1, y_2,…,y_k)$\in Y$.

In (1), x is called a decision variable vector and y is called an objective function vector. The decision variable space is denoted by X and the objective function space is

H. Deng et al. (Eds.): AICI 2011, CCIS 237, pp. 68–75, 2011.

denoted by Y. The constraint condition $e(x) \leq 0$ determines the set of feasible solutions. The set of solutions of multiobjective optimization problems consist of all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another [1]. Differently from Single-objective Optimization Problems (SOPs), multiobjective optimization problems have a set of solutions known as the Pareto optimal set. This solution set is generally called non-dominated solutions and is optimal in the sense that no other solutions are superior to them in the search space when all objectives are considered.

The set of objectives forms a space where points in the space represent individual solutions. The goal of course is to find the best or optimal solutions to the optimization problem at hand. Pareto optimality defines how to determine the set of optimal solutions. A solution is Pareto-optimal if no other solution can improve one objective function without a simultaneous deterioration of at least one of the other objectives. A set of such solutions is called the Pareto-optimal front.

Evolutionary algorithms (EA's) seem to be particularly suited for this task because they process a set of solutions in parallel, possibly exploiting similarities of solutions by recombination. Some researchers suggest that multiobjective search and optimization might be a problem area where EA's do better than other blind search strategies [2].

First, they can be applied when one has only limited knowledge about the problem being solved. Second, evolutionary computation is less susceptible to becoming trapped by local optima. This is because evolutionary algorithms maintain a population of alternative solutions and strike a balance between exploiting regions of the search space that have previously produced fit individuals and continuing to explore uncharted territory. Third, evolutionary computation can be applied in the context of noisy or non-stationary objective functions.

At the same time, difficulties can and do arise in applying the traditional computational models of evolution to multiobjective optimization problem. There are two primary reasons traditional evolutionary algorithms have difficulties with these types of problems. First, the population of individuals evolved by these algorithms has a strong tendency to converge because an increasing number of trials are allocated to observed regions of the solution space with above average fitness. This is a major disadvantage when solving multimodal function optimization problems where the solution needs to provide more information than the location of a single peak or valley [3]. This strong convergence property also precludes the long-term preservation of coadapted subcomponents required for solving covering problems or utilizing the divide-and-conquer strategy, because any but the strongest individual will ultimately be eliminated. Second, individuals evolved by traditional evolutionary algorithms typically represent complete solutions and are evaluated in isolation. Since interactions between population members are not modeled, even if population diversity were somehow preserved, the evolutionary model would have to be extended to enable coadaptive behavior to emerge [4].

To avoid this phenomenon, we proposed a multiobjective coevolutionary genetic algorithm (ECCA) for multiobjective optimization. Individual evolution is based on its fitness in genetic algorithm, but its living environment and relationship with other part aren't envolved. Coevolution is the process of mutual adaptation of two or more populations. The computational study of coevolution initiated by Hillis gave birth to competitive coevolutionary algorithms. His main motivation of the work reported here was precisely to take advantage of some coevolutionary concept [5]. In 1994, Paredis introduced Coevolutionary Genetic Algorithms (CGAs) [6]. In contrast with the typical all-at-once fitness evaluation of Genetic Algorithms (GAs), CGAs employ a partial but continuous fitness evaluation. Furthermore, the power of CGAs was demonstrated on various applications such as classification, process control, and constraint satisfaction. In addition to this, a number of symbiotic applications have been developed.

The major concern of this paper is to introduce the idea of competitive coevolution into multiobjective optimization. A multiobjective competitive coevolutionary genetic algorithm is proposed and implemented based on this idea to search the Pareto front effectively and reduce the runtime. At each generation, the proposed approach store the Pareto optimal point obtained along the evolutionary process into external set, enforcing a more uniform distribution of such vectors along the Pareto front. We then describe multiobjective optimization problems and a typical test functions that we use to judge the performance of the ECCA. Empirical results from the ECCA runs are presented and compared to previously published results.

The paper is organized as follows. Section 2 gives the ecological population competition mode and the proposed algorithm. In Section 3, he results of simulation are presented. The conclusions are given in Section 4.

2 The Proposed Algorithm

In collaborative evolution individual's self-status, living environment and competition with other individuals affect individual's self-evolution [7]. Lotka-Volterra competition equation as the theoretical model of population competition is introduced to describe populations' cooperation. Given two populations N_1, N_2, the cooperation between them can be formulated as follows (2) and (3):

$$\frac{dN_1}{dt} = r_1 N_1 \left(\frac{K_1 - N_1 - a_{12} N_2}{K_1} \right) \tag{2}$$

$$\frac{dN_2}{dt} = r_2 N_2 \left(\frac{K_2 - N_2 - a_{21} N_1}{K_2} \right) \tag{3}$$

Where K_1, K_2 are the living environment loads of population N_1, N_2 without competition to each other, and r_1, r_2 are individual's maximum increasing rates. a_{12}, a_{21} are competition coefficients, a_{ij} represents the suppression effect of individuals of population N_i from individuals of population N_j. In community comprising *n* different populations, competition equation can be formulated as follows (4):

$$\frac{dN_i}{dt} = r_i N_i \left(\frac{K_i - N_i - \left(\sum_{j=1}^{n} a_{ij} N_j \right)}{K_2} \right) \tag{4}$$

It's the cooperation model based on ecological population density. W e could exploit this model to describe the relation of multiple objectives. Because the relation of multiple objectives is just collaborative coexistence, and it finally is stable.

Exploiting (4), we proposed a ECCA algorithm based on ecological cooperation. The primary design goal of the proposed approach is to produce a reasonably good approximation of the true Pareto front of a problem. The proposed algorithm using dynamical equation of population competition at ecology to describe the complex, nonlinear relations of multiple objectives and to adjust the relation on individual and population levels simultaneously.

In our algorithms, each objective corresponds to a population. At each generation, these populations compete among themselves. An ecological population density competition equation is used for reference to describe the relation between multiple objectives and to direct the adjustment over the relation at individual and population levels. Moreover, t he proposed approach store the Pareto optimal point obtained along the evolutionary process into external set, enforcing a more uniform distribution of such vectors along the Pareto front.

The basic idea of algorithm is as follows:

1. Each objective corresponds to a population;
2. In one iterative step, evolution process and cooperation process must be executed; the evolutionary process adopts GA's genetic operations, while the cooperation process adopts (4) to compute population density and to adjust the scales of populations. The scale of population is formulated as (5):

$$N_i(t+1)=N_i(t)+dN_i/dt \tag{5}$$

If the increasing of population N_i is positive, randomly generated dN_i/dt individuals join population N_i for enlarging the scale of N_i.

If the increasing of population N_i is negative, according to the fitness of population N_i, dN_i/dt individuals with minimal fitness are deleted. The scale of population is reduced.

As a complete unit, ECCA pseudo code description is given here:

```
Step1: Initialize a null set as external set
Step2: for all objective functions f_i(x)
           Initialize a random population N_i corresponding to f_i(x)
       endfor
Step3: while (terminative condition is NOT satisfied)
           for all populations N_i
               (A) Compute all of the Pareto optimal points P_i in population N_i
               (B) Store P_i into external set
               (C) General genetic operations are performed
               (D) Computing dN/dt using (4)
               (E) Determine next generation's scale of population N_i using (5)
           endfor
   endwhile.
```

3 Results of Simulation

To validate our approach, we used the methodology normally adopted in the evolutionary multiobjective optimization literature [8]. We performed quantitative comparisons (adopting three metrics) with respect to three MOEAs that are representative of the state-of-the-art in the are: the microGA for multiobjective optimization [9], the Pareto Archived Evolution Strategy (PAES) [10] and the Nondominated Sorting Genetic Algorithm II (NSGA-II) [11]. For our comparative study, we implemented for three following metrics:

1. Spacing (SP): This metric was proposed by Schott [12] as a way of measuring the range (distance) variance of neighboring vectors in the Pareto front known. This metric is defined as (6):

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2} \qquad (6)$$

Where $d_i = \min_j (\Sigma_{k=1}^{m} \left| f_m^i - f \right|_m^j), i, j = 1,2,\ldots,n$, m is the number of objectives, $\bar{d}$ is the mean of all d_i, and n is the number of vectors in the Pareto front found by the algorithm being evaluated. A value of zero for this metric indicates all the nondominated solutions found are equidistantly spaced.

2. Generational Distance (GD): The concept of generational distance was introduced by Van Veldhuizen & Lamont [13] as a way of estimating how far are the elements in the Pareto front produced by our algorithm from those in the true Pareto front of the problem. This metric is defined as (7):

$$GD = \frac{\sqrt{\Sigma_{i=1}^{n} d_i^2}}{n} \qquad (7)$$

where n is the number of nondominated vectors found by the algorithm being analyzed and d_i is the Euclidean distance (measured in objective space) between each of these and the nearest member of the true Pareto front. It should be clear that a value of GD=0 indicates that all the elements generated are in the true Pareto front of the problem. Therefore, any other value will indicate how “far” we are from the global Pareto front of our problem..

3. Error Ratio (ER): This metric was proposed by Van Veldhuizen [14] to indicate the percentage of solutions (from the nondominated vectors found so far) that are not members of the true Pareto optimal set as (8):

$$ER = \frac{\Sigma_{i=1}^{n} e_i}{n} \qquad (8)$$

where n is the number of vectors in the current set of nondominated vectors available; e_i= 0 if vector i is a member of the Pareto optimal set, and e_i=1 otherwise. It should then be clear that ER=0 indicates an ideal behavior, since it would mean that all the vectors generated by our MEA belong to the Pareto optimal set of the problem. This metric addresses the third issue from the list previously provided.

We test the performance of ECCA on test function defined as follows (9):

$$
\begin{aligned}
&Minmize \quad f_1(x_1, x_2) = x_1 \\
&Minimize \quad f_2(x_1, x_2) = (1.0 + 10.0x) \cdot \\
&\left(1.0 - \frac{x_1}{1.0 + 10.0x} - \frac{x_1}{1.0 + 10.0x}\sin(\pi 4 x_1)\right) \\
&0.1 \le x_1, x_2 \le 1.0
\end{aligned}
\tag{9}
$$

To compute the nondominated front for the ECCA, we did the following. For each ECCA run, we collected all the Pareto optimal point into external nondominated set corresponding to the individuals evaluated during the run.

In this example, our approach used: $popsize_{init}$= 100, $popsize_{external}$ =30. Table 1 shows the values of the metrics for each of the MOEAs compared.

As noted in the literature [15], comparing multiobjective optimization algorithms against each other can be difficult. One would like an algorithm to minimize the distance to the Pareto optimal front and provide uniform coverage of the Pareto optimal front for a wide range of values. Thus, comparisons become multiobjective optimization problems themselves: is an algorithm that finds a handful of Pareto optimal solutions better than an algorithm that finds a wide, uniform distribution of near Pareto optimal solutions? With this in mind we present the experimental results according to different algorithms shown in Table 1.

Table 1. Results of Simulation

		CO-MOEA	MicroGA	PAES	NSGAII
ER	best	0.46	0.42	0.02	0.00
	median	0.61	0.77	0.07	0.02
	worst	0.68	0.98	0.15	0.08
	average	0.60	0.75	0.07	0.03
	std. dev.	0.061	0.145	0.030	0.021
GD	best	0.0003	0.0008	0.0001	0.0007
	median	0.001	0.0089	0.0006	0.0008
	worst	0.042	0.238	0.0659	0.0009
	average	0.0049	0.0681	0.0066	0.0008
	std.dev.	0.009	0.086	0.016	0.000
SP	best	0.006	0.017	0.007	0.006
	median	0.012	0.042	0.014	0.008
	worst	0.379	1.539	0.624	0.086
	average	0.039	0.356	0.054	0.01
	std.dev.	0.073	0.507	0.141	0.014

From Table 1, we can see that the NSGA-II had the best overall performance. It is also clear that the microGA presented the worst performance for this test function. Based on the values of the ER and SC metrics, we can conclude that our approach had problems to reach the true Pareto front of this problem. Note however, that the values of GD and SP indicate that our approach converged very closely to the true Pareto front and that it achieved a good distribution of solutions. PAES had a good performance regarding closeness to the true Pareto front, but its performance was not so good regarding uniform distribution of solutions.

4 Conclusion

The solution of multiobjective optimization problem is a set of indeterminacy points, and the task of multiobjective optimization is to estimate the distribution of this solution set, then to find the satisfying solution in it [16]. Many methods solving multiobjective optimization using genetic algorithm have been proposed in recent twenty years. But these approaches tend to work negatively, causing that the population converges to small number of solutions due to the random genetic drift [17]. To avoid this phenomenon, a competitive coevolutionary genetic algorithm (ECCA) for multiobjective optimization is proposed. The primary design goal of the proposed approach is to produce a reasonably good approximation of the true Pareto front of a problem. In the algorithms, each objective corresponds to a population. At each generation, these populations compete among themselves. An ecological population density competition equation is used for reference to describe the relation between multiple objectives and to direct the adjustment over the relation at individual and population levels.

The proposed approach was validated using typical test function taken from the specialized literature. Our comparative study showed that the proposed approach is competitive with respect three other algorithms that are representative of the state-of-the-art in evolutionary multiobjective optimization. More work, and more comparisons is need to determine the general properties of ECCA, and how they can be adapted or improved.

References

1. Zitzler, E.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans. on Evolutionary Computation 13(4), 257–271 (1999)
2. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 23(4), 1–16 (1995)
3. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2), 173–195 (2000)
4. Ficici, S.G.: Solution concepts in coevolutionary algorithms, Ph.d. dissertation, Brandeis University (May 2004)
5. Hillis, W.D.: Co-evolving parasites improve simulated evolution as an optimization procedure. In: Artificial Life II, pp. 313–324. Addison-Wesley, Reading (1991)
6. Paredis, J.: Coevolutionary computation. Artificial Life 21(4), 355–379 (1995)

7. Cao, X.-b., Li, J.-l., Wang, X.-f.: Research on Multiobjective Optimization based on Ecological cooperation. Journal of Software 12(4), 521–528 (2001)
8. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002); ISBN 0-3064-6762-3
9. Coello Coello, C.A., Toscano Pulido, G.: Multiobjective Optimization using a Micro-Genetic Algorithm. In: Spector, L., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 274–282. Morgan Kaufmann Publishers, San Francisco (2001)
10. Knowles, J.D., Corne, D.W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation 8(2), 149–172 (2000)
11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
12. Van Veldhuizen, D.A., Lamont, G.B.: On Measuring Multiobjective Evolutionary Algorithm Performance. In: 2000 Congress on Evolutionary Computation, vol. 1, pp. 204–211. IEEE Service Center, Piscataway (2000)
13. Van Veldhuizen, D.A., Lamont, G.B.: On Measuring Multiobjective Evolutionary Algorithm Performance. In: 2000 Congress on Evolutionary Computation, vol. 1, pp. 204–211. IEEE Service Center, Piscataway (2000)
14. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2), 173–195 (2000)
15. De Jong, E.D., Pollack, J.B.: Idea evaluation from coevolution. Evolutionary Computation 12(4), 25–28 (2004)
16. Xing, Z.-Y., Zhang, Y., Hou, Y.-L., Jia, L.-M.: On Generating Fuzzy Systems based on Pareto Multi-objective Cooperative Coevolutionary Algorithm. International Journal of Control, Automation, and Systems 5(4), 444–455 (2007)
17. Horn, J., Goldberg, D.: A Niched Pareto genetic algorithm for multiobjective optimization. In: Proc. of the First IEEE Conference on Evolutionary Computation, Piscataway, pp. 82–87 (1994)