The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Competitive Coevolutionary Algorithms can Solve Function Optimization Problems

Tatsuya Sato Takaya Arita
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
E-mail: sato@alife.cs.is.nagoya-u.ac.jp, arita@nagoya-u.jp

Abstract

Competitive coevolutionary algorithms (CCEA) have many advantages but their application range has been crucially limited. This study provides a simple, non-problem-specific framework to extend the range. The framework has two coevolving populations: that of candidate solutions and that of criteria, in which these population competitively coevolve with each other. The framework aims to move candidate solutions getting stuck in a local optimum by changing the fitness landscape dynamically. Moreover, the framework has a mechanism in order to establish and maintain proper arms race. We have conducted experiments on two function optimization problems: 1-dimensional function maximization problem and the Rastringin function minimization problem, in order to investigate the basic properties of the framework. The results of the experiments showed that CCEA achieves comparable performance with GA.

Keywords: Competitive Coevolutionary Algorithm, Genetic Algorithm, Evolutionary Computation, Function Optimization, Artificial Evolution

1 Introduction

Since D. Hillis extended the evolutionary computation paradigm by introducing coevolution [1] *et al.*, competitive coevolutionary algorithms (CCEAs) have attracted a lot of attention in the field of evolutionary algorithms (EAs). The framework of CCEA is similar to that of conventional EA except for fitness evaluation, in which the fitness of an individual is based on "competitions" with other individuals instead of an objective function. The difference between CCEA and EA in fitness evaluation have been believed to make a significant difference between them in applicable problems.

Conventional EA can be applied to the problems in which solutions can be evaluated absolutely, such as function optimization, combinatorial optimization. On the other hand, CCEA can find great solutions for problems which can be expressed by using competitive populations in GA framework including sorting-network design (sorting networks vs test cases) [1], cellular automaton density classification (CA rules vs initial states) [2] and the Nim game (first move vs passive move) [3]. It has been shown that CCEA can perform better than EA in this type of problems. However, the range of the problems to which CCEA can be applied has been crucially limited, which has prohibited its progress.

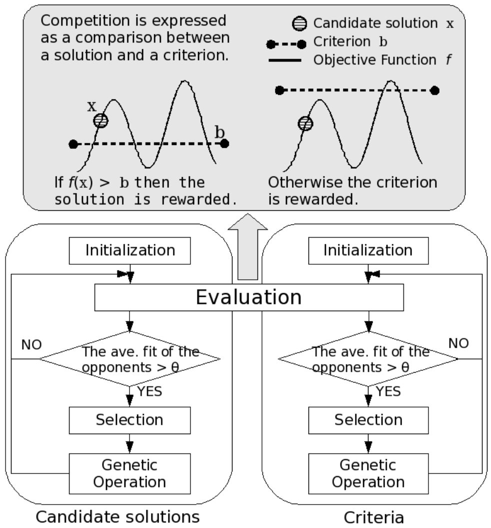


Figure 1: The framework.

There have been a very few researches which investigate the performance and characteristics of CEAs in solving problems in which solutions can be evaluated absolutely [4]. However, unfortunately approaches proposed in these researches are problem specific. In this work, we propose a simple, non-problem-specific framework of CCEA which extends significantly the range of problems to which CCEA can be applied.

2 Framework

Figure 1 shows our proposed framework. In the framework, candidate solutions and criteria (thresholds) competitively coevolve with each other. A criterion is defined as a real value used to evaluate a candidate solution. Competition between a solution and a criterion in a maximization problem is simply a comparison: If the objective function value of the candidate solution is greater than the criterion then the solution is rewarded, otherwise the criterion is rewarded. The total rewards are used as fitness. The fitness functions of solutions and criteria, denoted by f_{Sol} and f_{Crit} respectively, are defined as follows:

$$f_{\mathrm{Sol}}(x) = \sum_{b \in P_{\mathrm{Crit}}} I_{(\mathrm{Sol},\mathrm{Crit})}(x, b) \tag{1}$$

$$f_{\mathrm{Crit}}(b) = \sum_{x \in P_{\mathrm{Sol}}} I_{(\mathrm{Crit},\mathrm{Sol})}(b, x) \tag{2}$$

$$I_{(\mathrm{Sol},\mathrm{Crit})}(x, b) = \begin{cases} 1 & \text{if } f(x) \geq b \\ 0 & \text{else} \end{cases} \tag{3}$$

$$I_{(\mathrm{Crit},\mathrm{Sol})}(b, x) = 1 - I_{(\mathrm{Sol},\mathrm{Crit})}(x, b) \tag{4}$$

where $I_{(\mathrm{Sol},\mathrm{Crit})}(x, b)$ and $I_{(\mathrm{Crit},\mathrm{Sol})}(b, x)$ represent rewards to a candidate solution x and to a criterion b respectively, P_{Sol} and P_{Crit} represent the population of candidate solutions and that of criteria respectively and $f(x)$ is the objective function of a given problem.

The point of the framework is to utilize the loss of gradient of the dynamic landscape, which has been believed to be detrimental at least in the field of CCEA [5]. Specifically, the more one population outperforms the other, in other words, the less difference in fitness among individuals in each population exist, the more coevolutionary search becomes to random search, which is realized not by an explicit algorithm but by the coevolution implicitly. This mechanism can allow populations stuck in local optima to escape there.

Moreover, this framework has a mechanism in order to establish and maintain proper arms race. Selection and genetic operations are not performed to a population if the average fitness of the opponents is lower than the parameter θ. In other words, if the difference in fitness between two populations goes beyond a certain present value, the preceding population stops evolving until dropping below the value. Thus this framework stops the evolution of a coevolving population according to the condition of the opponents.

This framework is not problem-specific since it uses the objective function of a given problem only.

3 Evolutionary Setup

Our CCEA has two populations: a population of candidate solutions and that of criterion. Each population consists of N individuals. The genotype of a candidate solution is defined as a vector of length n and, as mentioned above, the genotype of a criterion is defined as a real value. For each of these, the genotype is identical to the phenotype. Stochastic Universal Selection [6] is adopted. Mutation is the only genetic operator used in the experiments, which is realized by adding a random number generated according to a normal distribution $\mathcal{N}(0, \sigma_{\mathrm{Sol}})$ for candidate solutions or $\mathcal{N}(0, \sigma_{\mathrm{Crit}})$ for criterion to the value at each locus with a probability p_m. All algorithms evaluated in this experiments terminates when the number of generations reached to $g_{\max}$.

Table 1 shows the values used for these parameters.

Table 1: The values used for parameters.

	1-D function	Rastringin
population size N	30	100
dimension n	1	10
mutation prob. p_m	0.1	0.01
σ_{Sol}	0.25	0.25
σ_{Crit}	0.25	1.0
max. generation $g_{\max}$	1×10^5	5×10^4

4 Results

In order to investigate the basic properties of the framework, we firstly applied it to a 1-dimensional function maximization problem and compared the results of out CCEA to those of three types of genetic algorithms (GAs): a conventional GA, a GA with fitness sharing (GA+FS) which is used to avoid premature convergence [7], a GA with random selection (RANDOM). Each of the algorithms compared is initialized with a population of N individuals whose genotype is 0.

The objective function of this problem is defined as a multimodal function of one variable as follows:

$$f(x) = x + ax\sin(bx) \tag{5}$$

where $x \in [0, \infty]$ and the values of a and b are 0.75 and 3.0 respectively. Local optima of this problem are placed at regular intervals.

Figure 2 shows that the objective value of the best solution which each algorithm found so far. CCEA found better solutions for this problem than GAs. Particularly, CCEA with $\theta = 0$ is the best among the algorithms compared in this experiments, which is 10^2 times as large as GA and is 10 times as large as GA+FS in terms of the objective value of the best solution found up to the last generation (Table 2).

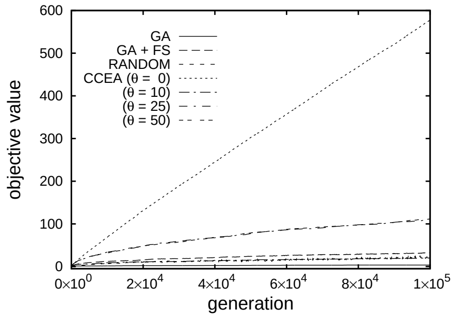


Figure 2: The objective value of the best solution found so far versus generation (averaged over 30 runs).

Table 2: The objective value of the best solution found up to the last generation (averaged over 30 runs).

GA		3.730×10^0
GA+FS		3.312×10^1
RANDOM		1.482×10^1
CCEA	$\theta = 0$	5.777×10^2
	$\theta = 10$	2.012×10^1
	$\theta = 25$	1.095×10^2
	$\theta = 50$	1.118×10^2

Figure 3 illustrates how in CCEA with $\theta = 0$ candidate solutions escape from local optima by utilizing loss of gradient. Candidate solutions can move and pass through valleys among local optima freely while loss of gradient occurs (Fig. 3(a)). The populations start to coevolve competitively again when a good solution whose objective value is larger than some criteria is found (Fig. 3(b)). Though loss of gradient will occur again, this phenomenon moves candidate solutions to a higher peak (Fig. 3(c)).

Furthermore, we applied our framework to the Rastringin function minimization problem [8] which is known as a benchmark problem so as to investigate how our framework would work on more practical problems. Rastringin function is given by the following equation:

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right) \qquad (6)$$

where $x_i \in [-5.12, 5.11]$. This problem has the unique global optimum $x^* = (0, 0, \ldots, 0)$ and has many local optima. A population of candidate solutions is initialized randomly and all criterion in the initial population are arranged at regular intervals in the range between the maximum of and the minimum of the objective function values of the initial candidate solutions.

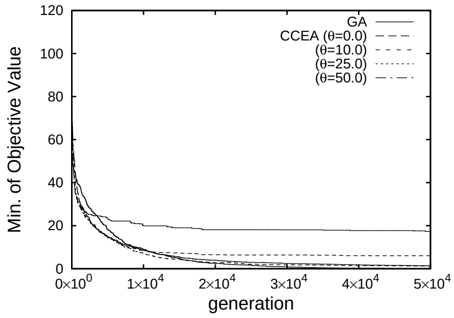


Figure 4: The objective value of the best solution found so far (averaged over 30 runs).

Table 3: The objective value of the best solution found so far (averaged over 30 runs).

GA		6.633×10^{-2}
CCEA	$\theta = 0.0$	1.726×10^1
	$\theta = 10.0$	6.641×10^0
	$\theta = 25.0$	1.923×10^0
	$\theta = 50.0$	1.025×10^0

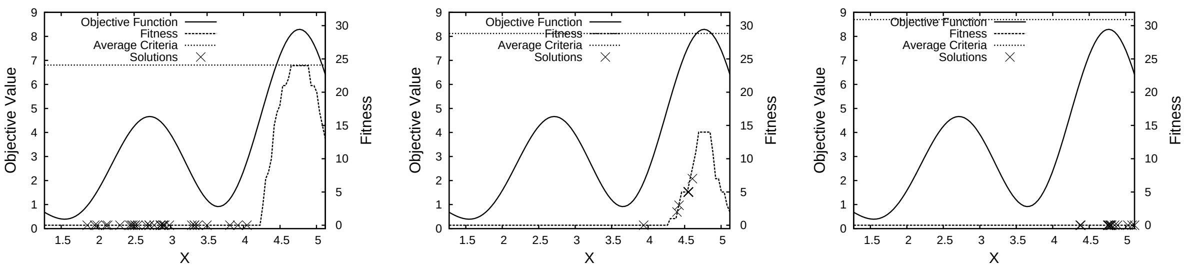
Figure 4 shows that the objective value of the best solution which each algorithm found so far. This figure indicates that the larger the value of the parameter θ, the better the quality of solutions which CCEA found. However, the performance of CCEA was slightly worse than that of GA (Table 3).

In order to measure the speed of convergence to the global optimum, we measured the expected number of generations taken by each algorithm to find a candidate solution whose objective value is less than a specified threshold (Figure 5)[1]. This analysis reveals that CCEA finds the nearest local optima ($n = 1$) to the global optimum faster than GA.

5 Summary

We have proposed a simple framework of competitive coevolutionary algorithms (CCEA) to extend their application range. We conducted the experiments on two function optimization problems: a 1-dimensional function maximization problem and the

[1] The values of multiplying the objective function value of the nearest local optimum to the global optimum by n ($0 \leq n \leq 10$) are used for the thresholds. In the figure, the values of all algorithms compared in the experiments at $n = 0$ are equal to 0, which indicates that all algorithm could not find the global optimum.



(a) 796th generation. (b) 802th generation. (c) 810th generation.

Figure 3: The changes of the dynamic fitness landscape in a typical run. The solid line indicates the objective function (Equation 5), the dashed line indicates the fitness landscape of candidate solutions and the dotted line indicates the average of criteria. A cross indicates a candidate solution.

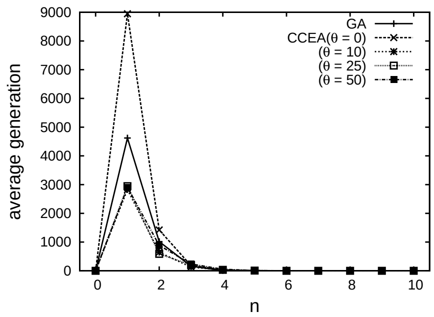


Figure 5: The expected number of generations taken by each algorithm to find a candidate solution whose objective value is less than a certain level. $n = 0$ means the global optimum (averaged over 30 runs).

Rastringin function minimization problem, in order to investigate the basic properties of our framework. The results of the first experiments showed that our CCEA is the best among the algorithms compared in the experiments because it can escape from local optima by utilizing loss of gradient and demonstrated how CCEA utilizes loss of gradient. Moreover, the results of the second experiments showed that CCEA has performance comparable to GA and that, in particular, on a practical problem our CCEA can find an approximate solution faster than GA.

Function optimization is one of the most important optimization problems and has a large application. We believe that this framework will open up an interesting possibility to extend drastically the range of problems to which CCEA can be applied.

Acknowledgments

This work was supported by a Grant-in-Aid for 21st Century COE "Frontiers of Computational Science".

References

[1] W. D. Hillis: "Co-evolving parasites improve simulated evolution as an optimization procedure", *Artificial Life II*, pp. 313–324, 1992.

[2] H. Juillé and J. B. Pollack: "Coevolutionary learning: a case study", *Proc. of the Fifteenth International Conference on Machine Learning*, pp. 251–259, 1998.

[3] C. D. Rosin: "Coevolutionary Search Among Adversaries", *PhD thesis*, University of California, San Diego, CA, 1997.

[4] J. Paredis: "Co-evolutionary constraint satisfication", *Proc. of the International Conference on Evolutionary Computation*, pp. 46–55, 1994.

[5] J. Cartlidge and S. Bullock: "Combating coevolutionary disengagement by reducing parasite virulence", *Evolutionary Computation*, Vol. 12, No. 2, pp. 193–222, 2004.

[6] J. E. Baker: "Reducing bias and inefficiency in the selection algorithm", *Proc. of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pp. 14–21, 1987.

[7] D. E. Goldberg and J. Richardson: "Genetic algorithms with sharing for multimodal function optimization", *Proc. of the Second International Conference on Genetic Algorithms*, pp. 41–49, 1987.

[8] D. Whiteley, S. B. Rana, J. Dzubera and K. E. Mathias: "Evaluating evolutionary algorithms", *Artificial Intelligence*, Vol. 85, Issue 1-2, pp. 245–276, 1996.