The Sixteenth International Symposium on Artificial Life and Robotics 2011 (AROB 16th '11),
B-Con Plaza, Beppu,Oita, Japan, January 27-29, 2011

An Implementation of Probabilistic Model-Building Coevolutionary Algorithm

Takahiro OTANI and Takaya ARITA

Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
(Email: t-otani@alife.cs.is.nagoya-u.ac.jp, arita@nagoya-u.jp)

Abstract: We propose an extended coevolutionary algorithm (CA) with probabilistic model-building (CA-PMB) in order to improve search performance of CA. This paper specifically describes an implementation of CA-PMB called coevolutionary algorithm with population-based incremental learning (CA-PBIL), and analyzes behavior of the algorithm through computational experiments using intransitive numbers game as a benchmark problem. The experimental results show that desirable coevolution may be inhibited by the *over-specialization* effect, and that the algorithm shows complex dynamics caused by the game's *intransitivity*. However, further experiments show that the intransitivity encourages desirable coevolution when a different learning rate is set for each population.

Keywords: Coevolutionary algorithm, Probabilistic model-building genetic algorithm, Intransitive numbers game**.**

I. INTRODUCTION

Competitive coevolutionary algorithm is an extension of standard evolutionary algorithms in which each solution is not evaluated by a fixed objective function (or fitness function), but is evaluated based on interactions between other solutions.

The first application of the algorithm is design of a sorting network proposed by Hillis [1]. He showed CA can design a feasible network structure which has fewer comparators than a network designed by a standard genetic algorithm. CA has also been successfully applied to various problems including evolution of artificial neural network for classification problems, function approximation, evolving strategies of game players, coevolving predator-prey robots, and so on.

In this study, we propose an extended CA called coevolutionary algorithms with probabilistic model-building (CA-PMB) in which probabilistic model-building genetic algorithm (PMBGA) [2] is used as a search heuristics, and analyzes behavior of the algorithm through computational experiments.

Standard CA uses the search mechanism of the genetic algorithm to evolve candidate solutions. CA-PMB adopts PMBGA as a search heuristics in order to improve search performance of standard CA. PMBGA (also called estimation of distribution algorithm) is a generic name given to a class of evolutionary algorithms in which genetic operators, crossover and mutation, are replaced by building a probabilistic model which represents the distribution of promising solutions and generating new solutions based on the model. PMBGA can explicitly deal with dependence of variables in a problem, and therefore it shows better search performance on various optimization problems than standard GAs. This paper shows an implementation of CA-PMB called coevolutionary algorithm with population-based incremental learning (CA-PBIL).

This paper analyzes behavior of CA-PBIL through computational experiments using intransitive numbers game as a benchmark problem. Intransitive numbers game is an abstract model devised by Watson and Pollack [3] to study *intransitivity* which refers to cyclic dominance such as the Rock-Paper-Scissors game, and has been seen as a substantial obstacle to progress in competitive coevolutionary algorithms. Experimental results show that desirable coevolution may be inhibited by the *over-specialization* effect, and that the algorithm shows complex dynamics caused by the game's intransitivity. However, further experiments show that the intransitivity encourages desirable coevolution when a different learning rate is set for each population.

II. COEVOLUTIONARY ALGORITHM

CA is an extension of standard evolutionary algorithms. An important difference between (competitive) CA and standard evolutionary algorithms is how to define an evaluation function. In CA, each solution is not evaluated by a fixed objective function as in the standard evolutionary computations, but is evaluated based on interactions between other solutions. A model of CA dealt in this paper uses two solution sets S^1 and S^2. A solution s in a solution set is evaluated based on interactions with all of solutions in the other solution set. An evaluation function $F(s)$ is defined as follows,

$$F(s)=\begin{cases}\sum_{t\in S^2} E(s,t) & \text{if } s\in S^1\\ \sum_{t\in S^1} E(s,t) & \text{if } s\in S^2\end{cases}, \qquad (1)$$

where $E(s,t)$ denotes a payoff of solution s depending on an interaction with solution t, which is defined based on a targeted problem.

III. COEVOLUTIONARY ALGORITHM WITH PROBABILISTIC MODEL-BUILDING

1. Scheme of Proposed Algorithm

PMBGA (also called estimation of distribution algorithm) is a generic name given to a class of evolutionary algorithms in which genetic operators, crossover and mutation, are replaced by building a probabilistic model which represents distribution of promising solutions and generating new solutions based on the model.

In this study, we propose coevolutionary algorithms with probabilistic model-building (CA-PMB) in which PMBGA is used as a search heuristics. A scheme of the algorithm is described in Fig. 1. This algorithm generates each solution set using probabilistic model which represents the distribution of promising solutions, and evaluates each generated solution according to the equation (1). Then, the model is updated based on good solutions.

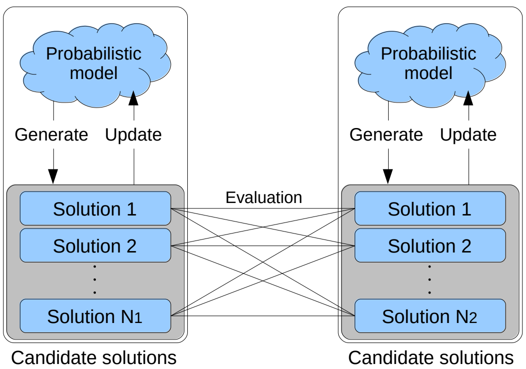


Fig. 1. Scheme of CA-PMB.

2. CA-PBIL: An Implementation of CA-PMB

This paper shows an implementation of CA-PMB called coevolutionary algorithm with population-based incremental learning (CA-PBIL). A pseudocode of the algorithm is described in Fig. 2. CA-PBIL uses PBIL algorithm [4] as a search heuristics. PBIL is a simple variation of PMBGA. This algorithm deals with candidate solutions as bit-strings, and uses and updates probability vector as a probabilistic model which represents the probability of assigning value "1" to the corresponding bit for a new candidate solution.

```
p_i^1 ← 0.5, i = 1,...,L_1
p_i^2 ← 0.5, i = 1,...,L_2
while Termination condition not met do
  S^1 ← N_1 solutions generated by p^1
  S^2 ← N_2 solutions generated by p^2
  best^1 ← argmax_{s∈S^1} F(s)
  best^2 ← argmax_{s∈S^2} F(s)
  p_i^1 ← p_i^1 · (1.0 − LR_1) + best_i^1 · LR_1, i = 1,...,L_1
  p_i^2 ← p_i^2 · (1.0 − LR_2) + best_i^2 · LR_2, i = 1,...,L_2
  for i = 1,...,L_1 do
    if rand < MP_1 then
      p_i^1 ← p_i^1 · (1.0 − MS_1) + rand(0,1) · MS_1
    end if
  end for
  for i = 1,...,L_2 do
    if rand < MP_2 then
      p_i^2 ← p_i^2 · (1.0 − MS_2) + rand(0,1) · MS_2
    end if
  end for
end while
```

Fig. 2. Pseudocode of CA-PBIL.

IV. INTRANSITIVE NUMBERS GAME

This paper analyzes the behavior of CA-PBIL through computational experiments using intransitive numbers game (ING) [3] as a benchmark problem. ING is an abstract model to study *intransitivity* (as in the well-known Rock-Paper-Scissors game) which has been seen as a substantial obstacle to progress in competitive coevolutionary algorithms, and is usually used to compare the performance of coevolutionary algorithms.

1. Problem Definition

A solution of ING is a n-dimensional vector, and a range of each dimension's value is $[0,k]$. The payoff function $E(s,t)$ is defined as follows,

$$E(s,t)=sign\left(\sum_{i=1}^{n} g_i\right), \qquad (2)$$

where

$$g_i=\begin{cases} s_i-t_i & \text{if } h_i=\min_j h_j\\ 0 & \text{otherwise}\end{cases}$$

$$h_i=|s_i-t_i|.$$

The function $sign$ returns 1 when an input value is positive value, returns -1 when an input value is negative value, and returns 0 when an input value is 0. Therefore, the winning solution is the one with higher magnitude in the dimension of least difference between the two solutions.

Furthermore, a modified ING is also used for experimental analysis of the algorithm. In this problem, the function $sign$ in equation (2) is modified as follows. The function returns 1 when an input value is positive value, otherwise returns 0. These settings cause algorithm's complex dynamics more frequently.

2. Bit-strings Solution Representation

A bit-string b is converted to a vector solution s as follows. The length of the bit-string is set to $L_1 = L_2 = nk$, and i-th dimension's value of a vector solution is calculated as follows,

$$s_i = \sum_{j=1}^{k} b_{(i-1)k+j} ,$$

where b_i denotes i-th bit value of the bit-string.

V. COMPUTATIONAL EXPERIMENTS

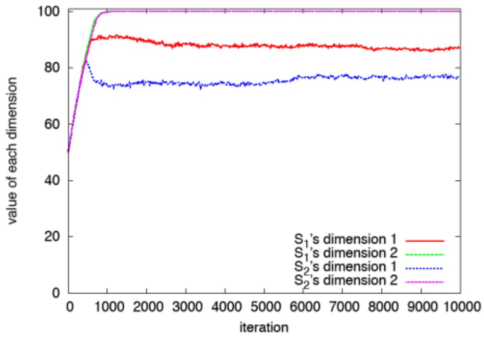
In this section, we analyze the behavior of CA-PBIL using ING. The number of dimension and the maximum value of each dimension were set to $n = 2$ and $k = 100$ respectively. Optimal solution is $(100,100)$. The length of a bit-string was set to $L_1 = L_2 = 200$.

1. Dynamics on the Standard ING

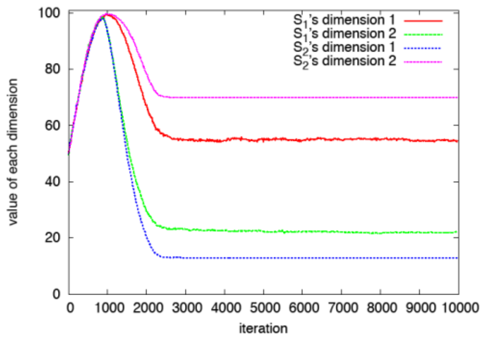
First of all, we carried out following experiments using the standard ING to study how the average evaluated value of generated solution progresses. Algorithm's parameters were set as follows, the learning rates: $LR_1 = LR_2 = 0.01$, the numbers of solutions generated every iteration: $N_1 = N_2 = 100$, and the mutation probabilities: $MP_1 = MP_2 = 0.0$. This means mutation procedure was not executed. The number of iteration was set at 10000, and the experiment was trialed 100 times.

The experimental results show that a *stagnation* pattern occurred in almost all trials while a *convergence to a low value* pattern rarely occurred. Typical examples of each occurred pattern are described in Fig. 3. Fig. 3(a) describes the stagnation pattern which is called the *over-specialization* effect [3]. If two values in one dimension become closer to each other than in the other dimension enough, the other dimension does not play a role in decision of payoffs. Therefore, the former will evolve desirably, however the latter will stagnate.

Fig. 3(b) shows a rare case in which the convergence to a low value pattern occurred. This pattern is caused by the intransitivity of the game, and is called *relativism* effect in the literature [3]. For example, consider $a = (4,7)$ and $b = (5,5)$. The dimension of least difference is the first, and b gets positive payoff. Now, $a' = (3,6)$ is a small variation from a. The dimension of least difference when a' and b is the second dimension and a' gets a positive payoff. Therefore, a should be replaced by a'. Then, $b' = (4,4)$ gets a positive payoff depending on the interaction with a', and b should be replaced by b'. These solutions evolve to low values in this manner repeatedly.



(a) Stagnation (99 times).



(b) Convergence to a low value (1 time).

Fig. 3. Typical examples of progress patterns on the standard intransitive numbers game.

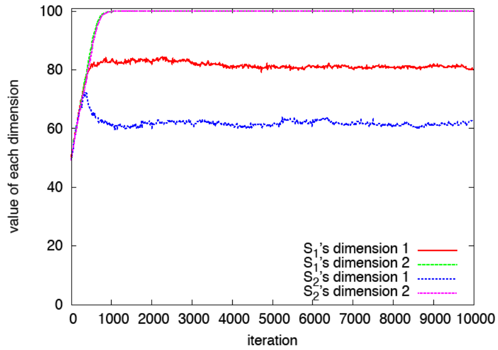
2. Dynamics on the Modified ING

We carried out the same experiments using the modified ING. Algorithm's parameters were set to same as previous experiments. In this experiments, four types of patterns occurred. Typical examples of each pattern are described in Fig. 4. Dynamic patterns described in Fig. 4(b)(c)(d) were caused by the intransitivity of the game.

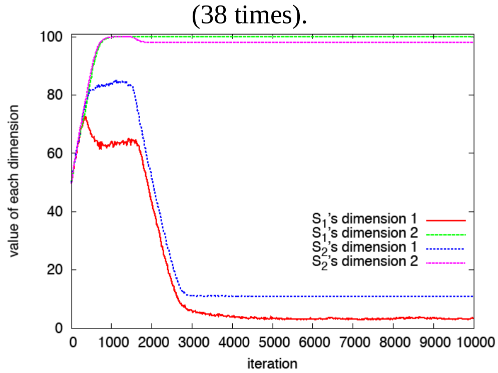
3. Coevolution under a Different Learning Rate

Finally, we carried out experiments to analyze the behavior of the algorithm when a different learning rate was set for each solution set. We examined the number of reached optimal solutions varying S^2's learning rate LR_2 from 0.01 to 0.03. And other parameters were set to same as the previous experiments.

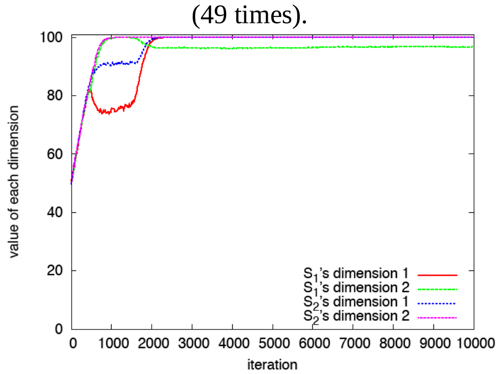
Fig. 5 shows the experimental results. According to these results, the search performance was increased when a different learning rate was set for each population on the modified ING. The reason is supposed to be that these settings could encourage convergence to high values as shown in Fig. 4(c).



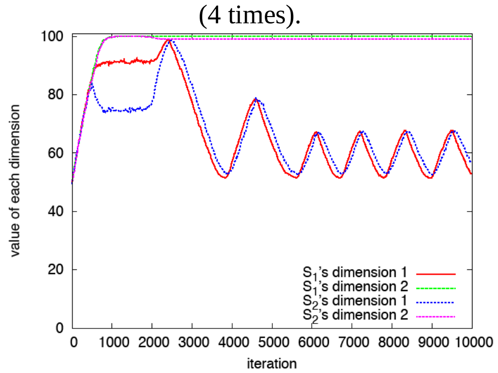
(a) Stagnation caused by the over-specialization (38 times).



(b) Converges to a low value in one dimension (49 times).



(c) Convergence to high values in both dimensions (4 times).



(d) Oscillation starting in one dimension (9 times).

Fig. 4. Typical examples of progress patterns on the modified intransitive numbers game.

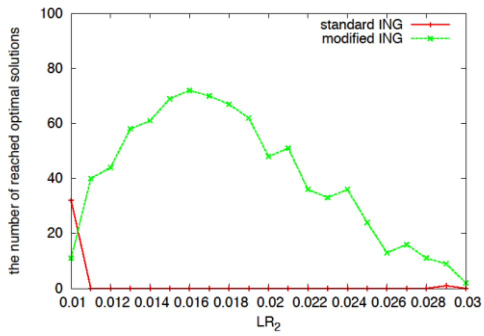


Fig. 5. The number of reached optimal solutions.

VI. CONCLUSION

In this paper, we proposed an extended CA called CA-PMB, showed CA-PBIL as an initial implementation, and examined the algorithm's behavior through experiments using ING. Experimental results showed that desirable coevolution may be inhibited by over-specialization effect. We also observed an interesting coevolutionary behavior caused by the game's intransitivity especially on the modified ING. Further experiments showed that the intransitivity encourages desirable coevolution when a different learning rate is set for each population. This feature has a possibility to provide new ways to solve various problems in coevolutionary domain more effectively and efficiently.

Future work includes the analysis of the proposed algorithm by conducting further experiments. In particular, the effectiveness of a different learning rate should be evaluated in more detail. Another direction would be to compare the performance with other algorithms and to evaluate using other more practical problems. It might be also interesting to implement CA combined with other PMBGA.

REFERENCES

[1] W. D. Hillis (1990), Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena* 42:228-234

[2] M. Pelikan, D. E. Goldberg, and F. G. Lobo (2002), A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications* 21:5-20

[3] R.A. Watson and J.B. Pollack (2001), Coevolutionary dynamics in a minimal substrate. *Proc. of the Genetic and Evolutionary Computation Conference*:702-709

[4] S. Baluja (1994), Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. *Technical Report* CMU-CS-94-163, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA