Adaptation to a dynamic environment by means of the environment identifying genetic algorithm

Conference Paper · February 2000

DOI: 10.1109/IECON.2000.972467 · Source: IEEE Xplore

Keinosuke Matsumoto
Osaka Prefecture University
71 PUBLICATIONS **579** CITATIONS

Adaptation to a Dynamic Environment by Means of the Environment Identifying Genetic Algorithm

Naoki MORI mori@cs.osakafu-u.ac.jp
Toshihiro KUDE kude@ss.cs.osakafu-u.ac.jp
Keinosuke MATSUMOTO matsu@cs.osakafu-u.ac.jp
College of Engineering Osaka Prefecture University
Sakai, Osaka 599-8531, JAPAN

Abstract

Adaptation to dynamic environments is an important application of genetic algorithms (GAs). However, there are many difficulties to apply the GA to dynamic environments. Especially, in online environments, the GA's defects become remarkable because individuals should be evaluated in the real world.

In this paper, we proposes a novel approach to such an online adaptation called the environment identifying Genetic Algorithm (EIGA). The EIGA achieves the online adaptation and identification of the environments simultaneously by the parallel technique and reduce the number of fitness evaluations in the real world by utilizing the identified environment. The thermodynamical selection rule is also utilized to maintain diversity.

Computer simulation is carried out by taking an Nk-landscape problem as an example.

1 Introduction

The Genetic Algorithms (GAs) is a search and optimization technique[1, 2] based on the mechanism of evolution by natural selection. Adaptation to dynamic environments is one of the most important applications of the GA. For example, in job scheduling in production systems, re-scheduling of the jobs is often required due to change of the specification of jobs or malfunction of the machines. Industrial products such as elevator systems, are required to adapt to customers whose behavior cannot be estimated well in advance. GAs are expected to be applicable to such problems as adaptation techniques.

Several studies have been reported on GAs for solving dynamic optimization problems. J. J. Grefenstette[3], H. G. Cobb and J. J. Grefenstette[4] have been proposed methods of controlling the mutation rate. N. Mori has proposed to utilize the thermodynamical genetic algorithm (TDGA)[5] to such problems[6, 7]. The TDGA is a genetic algorithm which evaluates the diversity of the population by *the entropy*, and selects the population so as to minimize *the free energy*[5]. If the environmental change is recurrent, memorizing the results of the past adaptations and utilizing them as candidates of the solutions will be an effective strategy. In the context of GA research, several studies in this approach have been proposed[2, 8]. In these studies, results of the past adaptation are memorized by introducing redundant genetic representation such as diploidy. The memory based TDGA also has been proposed[9].

To apply GAs to the real applications, the online adaptation and the offline adaptation should be considered. If the adaptation is achieved by a trial-and-error method directly committing the real world, we call this approach *the online adaptation*.

On the other hand, if the dynamic environment is identified at the beginning, and then, adaptation scheme is computed based on the identified problem, we call this approach *the offline adaptation*.

If an environment can be identified, the offline adaptation becomes an effective way. While if an environment is hard to be identified, the online adaptation is only solution. In the online adaptation, there are following difficulties. First one is that the possible number of fitness evaluations is limited. Therefore optimization must be achieved within the available number of evaluations. Second one is that to avoid extreme performance degradation by trial-and-error is required. To evaluate individuals which effect fatal damage for the real system also should be avoided.

Generally, the online adaptation is more difficult for GAs than the offline adaptation. Kamihira et al. have proposed *the Bio Control Architecture* which achieves the online optimization of an engine controller by GA[10]. Vavak et al. have proposed to apply the GA to the online load balancing of sugar beet presses[11].

However many of the real applications are hard to be solved by GAs only the online adaptation. For example, if the precision of identification is poor, the hybrid approach of using both the online and offline adaptation is required. In this paper, we propose a novel method called *the Environment Identifying Genetic Algorithm* (EIGA)[12], which can adapt the dynamic environments as identifying the environment. To execute the adaptation and the identification in parallel, *the Island Thermodynamical Genetic Algorithm* (ITDGA)[13] is utilized in the EIGA. The ITDGA is one of the parallel genetic algorithm which combines the TDGA with the island model GA[14, 15]. The performance and the adaptation ability of proposed method are confirmed by computer simulation taking an Nk-landscape model as examples[16, 17].

2 Island Model Thermodynamical Genetic Algorithm(ITDGA)

2.1 Thermodynamical Genetic Algorithm

In the TDGA, the selection operation is designed to minimize the free energy of the population. The free energy F is defined by:

$$F = \langle E \rangle - HT, \tag{1}$$

where $\langle E \rangle$ is the mean energy of the system, H is the entropy and T is a parameter called the temperature. Minimization of the free energy can be interpreted as taking a balance between minimization of the energy function and maintenance of the diversity measured by the entropy. The diversity of the population can be controlled by adjusting the temperature T explicitly.

2.2 Island Model Genetic Algorithm

The island model genetic algorithm (IGA)[14, 15] is one of the parallel genetic algorithms.

In IGA, population is divided into several subpopulations, each of which searches an independent sequential GA on its own subpopulation. This subpopulation is called *the island*. Occasionally, the elite individual of each island is transported to other islands. This operation is called *migration*.

2.3 Island Model Thermodynamical Genetic Algorithm

The island model thermodynamical genetic algorithm (ITDGA)[13] is the parallel genetic algorithm which combines the TDGA with the IGA.

ITDGA divides the population into several subpopulations just like IGA. The difference between ITDGA and IGA is the GA utilized in each island is replaced Simple GA[2] to TDGA in ITDGA.

3 Environment Identifying Genetic Algorithm(EIGA)

The outline of EIGA is as follows. The EIGA is consists of two main parts called *the online search module* and *the offline search module*. Both modules have the GA population.

The online search module identifies the environment and evaluates the individuals in the real world. The identification of the environment is achieved by the neural network[18]. The input vector of the network is the genotype of the individual in the offline search module and the output of the network is the fitness value of the input individual. In the online search module, the individual represents the set of weight values of the neural network and the real-coded GA is executed. We call the individual in the online search module *the environment individual* and the population *the environment population.*

On the other hand, the offline search module finds the new individuals by using the identified environments which are found in the online search module. There are several islands in the offline search module. We call the individual in the offline search module *the solution individual* and the population *the solution population.*

The ITDGA of the star structure[12] where the online search module is the center and islands of the offline search module are clients is utilized in the EIGA so as to try various identified environments in each island at the same time. This design is indispensable in the dynamic environment because the system is hard to follow the environmental change if there are only one identified environment, especially in the case that the precision of identification is poor.

In the offline search module, there are several islands and each island has its own identified environments. Therefore the fitness function is different in each island. One superior solution individual of each island is sent to the online search module at regular intervals. Only these solution individuals are applied to the real world. The online search module has several neural networks as environment individuals. The fitness value of the environment individual is calculated by the difference between the real fitness value and the output value of the neural network of environment individual as follows:

$$f = -\frac{\sum_{i=0}^{n-1} |F_{\mathrm{true}}(i) - F(i)|}{n} \qquad (2)$$

where $F_{\mathrm{true}}(i)$ is the real fitness value of i-th solution individual in the training data, $F(i)$ is the output value obtained by the neural network of the environment individual and n represents the number of the training data. The training data are solution individuals from the offline search module.

3.1 Online Search Module

The Real-Coded GA with the unimodal normal distribution crossover (UNDX)[19] is utilized in the online search module. The following mutation operator is also applied to each locus according to a prescribed mutation rate.

1. Let $i = 1$.
2. Select locus i.
3. Let $r^i = N(0, 1)$, where $N(0, 1)$ is normally distributed random number.
4. if $r^i > 0$, let $r^i = (g^i_{\mathrm{max}} - g^i_{\mathrm{old}}) \times r^i/3$, otherwise let $r^i = (g^i_{\mathrm{old}} - g^i_{\mathrm{min}}) \times r^i/3$.
5. Let $g^i_{\mathrm{new}} = g^i_{\mathrm{old}} + r$. If $g^i_{\mathrm{new}} > g^i_{\mathrm{max}}$ or $g^i_{\mathrm{new}} < g^i_{\mathrm{min}}$, go to Step 3.
6. If $i <$ chromosome length, let $i = i + 1$ and go to Step 2, otherwise terminate the mutation.

where g^i_{old} represents the value of the gene on locus i before the mutation, g^i_{new} represents the value of the locus i after the mutation, g^i_{max}, g^i_{min} represent the maximum range and the minimum range of the value of the gene on locus i respectively.

To maintain diversity, environment individuals within certain Euclid distance between the already selected environment individuals are removed from the target of selection. This distance called *the exclusive radius*[20] is set before the search. The individual is selected one by one in order of the fitness value under the condition of the exclusive radius.

The algorithm of the online search module at the generation t is as follows.

1. Set the exclusive radius R and the environment population size N_{E}.
2. Preserve the individual having the best fitness value as an elite e.
3. Pair all the individuals in environment population $\mathcal{P}_{\mathrm{E}}(t)$ randomly, and apply the UNDX operator to all the pairs, and obtain N_{E} offsprings $\mathcal{P}_{\mathrm{Eo}}(t)$. Apply the mutation operator to all the N_{E} parents and obtain $\mathcal{P}_{\mathrm{Ep}}(t)$. Let $\mathcal{P}'_{\mathrm{E}}(t) = \mathcal{P}_{\mathrm{Eo}}(t) \cup \mathcal{P}_{\mathrm{Ep}}(t) \cup e$.
4. Remove the overlapping individuals in $\mathcal{P}'_{\mathrm{E}}(t)$. If the $|\mathcal{P}'_{\mathrm{E}}(t)| < N_{\mathrm{E}}$, add the $N_{\mathrm{E}} - |\mathcal{P}'_{\mathrm{E}}(t)|$ random environment individuals to $\mathcal{P}'_{\mathrm{E}}(t)$, and let $\mathcal{P}_{\mathrm{E}}(t+1) = \mathcal{P}'_{\mathrm{E}}(t)$ and go to Step 10.
5. Let $i = 0$, $\mathcal{P}_{\mathrm{E}}(t + 1) = \mathcal{P}_{\mathrm{tmp}}(t + 1) = \phi$ and $r = R$.
6. If $\mathcal{P}'_{\mathrm{E}}(t) = \phi$, move all the individuals in $\mathcal{P}_{\mathrm{tmp}}(t+1)$ to $\mathcal{P}'_{\mathrm{E}}(t)$ and let $r = r \times 0.9$.
7. Select the individual h which has the best fitness value in $\mathcal{P}'_{\mathrm{E}}(t)$ and move h from $\mathcal{P}'_{\mathrm{E}}(t)$ to $\mathcal{P}_{\mathrm{E}}(t+1)$.
8. The individuals in $\mathcal{P}'_{\mathrm{E}}(t)$ which distance to h is within r are moved from $\mathcal{P}'_{\mathrm{E}}(t)$ to $\mathcal{P}_{\mathrm{tmp}}(t + 1)$.
9. If $i < N_{\mathrm{E}}$, $i = i + 1$ and go to Step 6. Otherwise go to Step 10.
10. The neural network of the elite individual in $\mathcal{P}_{\mathrm{E}}(t + 1)$ is trained by the backpropagation[18] and terminate the algorithm.

3.2 Offline Search Module

The algorithm of the island i at the generation t is as follows.

$N_{\mathrm{S}i}$ represents the subpopulation size of the island i which is given as N_{S}/I, where N_{S} is the solution population size and I is the number of islands in the offline search module.

1. Preserve the individual having the best fitness value as an elite e.
2. Pair all the individuals in the solution population $\mathcal{P}_{\mathrm{S}i}(t)$ randomly, and apply the crossover operator to all the pairs and obtain $N_{\mathrm{S}i}$ offsprings.
3. Apply the mutation operator to all the $N_{\mathrm{S}i}$ parents and $N_{\mathrm{S}i}$ offsprings and obtain $\mathcal{P}'_{\mathrm{S}i}(t)$.
4. Apply the thermodynamical selection of TDGA[5] to $\mathcal{P}'_{\mathrm{S}i}(t) \cup e$ and obtain $\mathcal{P}_{\mathrm{S}i}(t + 1)$.

In the offline search module, the migration is not carried out because the fitness value is given by the identified environments are different in each island.

3.3 Algorithm of Migration

In EIGA, the solution individuals found in the offline search module migrate to the online search module so as to be applied to the real world. Applying too many solutions and poor solutions to the real world is inadequate, only one superior solution individual is selected for the immigrant individual in each island. The selected individual is not always the elite because the solution individual is selected not to overlap the other island selected individual.

The algorithm of the migration from the offline search module to the online search module at the generation t is as follows.

$\mathcal{I}_i(t)$ represents an immigrant individual of the island i at the generation t.

1. Let immigrant population $\mathcal{P}_{\mathrm{M}}(t) = \phi$ and $i = 0$.
2. Let $j = 1$.
3. If $j > 3$ go to Step 5. Otherwise, let $\mathcal{I}_i(t)$ be the j-th best solution individual in the island i.
4. If there is no identical individual to $\mathcal{I}_i(t)$ in $\mathcal{P}_{\mathrm{M}}(t)$, add $\mathcal{I}_i(t)$ to $\mathcal{P}_{\mathrm{M}}(t)$ and go to Step 5. Otherwise, $j = j + 1$ and go to Step 3.
5. Let $i = i + 1$. If $i < I$, go to Step 2. Otherwise, go to Step 6.
6. Apply all the immigrant individuals in $\mathcal{P}_{\mathrm{M}}(t)$ to the real environment and get its real fitness value.
7. The immigrant individuals in $\mathcal{P}_{\mathrm{M}}(t)$ are added to the training data set of the online search module $\mathcal{P}_{\mathrm{L}}(t)$. The overlapping individuals in $\mathcal{P}_{\mathrm{L}}(t)$ with $\mathcal{P}_{\mathrm{M}}(t)$ are removed from $\mathcal{P}_{\mathrm{L}}(t)$, and let $\mathcal{P}'_{\mathrm{L}}(t) = \mathcal{P}_{\mathrm{L}}(t) \cup \mathcal{P}_{\mathrm{M}}(t)$. If $|\mathcal{P}'_{\mathrm{L}}(t)| > N_{\mathrm{L}}$, remove $N_{\mathrm{L}} - |\mathcal{P}'_{\mathrm{L}}(t)|$ oldest individuals in $\mathcal{P}'_{\mathrm{L}}(t)$ and let $\mathcal{P}_{\mathrm{L}}(t+1) = \mathcal{P}'_{\mathrm{L}}(t)$, where N_{L} is the number of the training data.

4 Algorithm of EIGA

The algorithm of EIGA is as follows.

1. Select appropriate values for
   N_{S}: the solution population size,
   N_{E}: the environment population size,
   N_{L}: the maximum number of the training data,
   N_{g}: the maximum number of generations, and
   I: the number of the islands in the offline search module.
2. Let $t = 0$.
3. Construct the initial environment population $\mathcal{P}_{\mathrm{E}}(0)$ randomly.
4. Construct the initial solution populations of all islands randomly. The solution population of island i is represented by $\mathcal{P}_{\mathrm{S}i}(0)$.
5. Construct the initial training data $\mathcal{P}_{\mathrm{L}}(0)$ with one solution individual randomly selected from each islands.
6. Apply all training data in $\mathcal{P}_{\mathrm{L}}(0)$ to the real world and obtain their real fitness values.
7. Calculate the fitness value of all the environment individuals of $\mathcal{P}_{\mathrm{E}}(0)$ by utilizing the training data in $\mathcal{P}_{\mathrm{L}}(0)$.
8. Select the higher rank I environment individuals in $\mathcal{P}_{\mathrm{E}}(0)$ and set these environment individuals as $E_i(0)$, where $E_i(t)$ represents the identified environment of island i at the generation t.
9. Let $i = 0$.
10. Search the solution in $\mathcal{P}_{\mathrm{S}i}(t)$ for one generation according to the algorithm of **3.2**. TDGA is utilized in each island.
11. Let $i = i + 1$. If $i < I$, go to Step 10.
12. Add the immigrant individuals of the offline search module to the training data set of the online search module according to the algorithm of **3.3**.
13. Search the solution in $\mathcal{P}_{\mathrm{E}}(t)$ for one generation according to the algorithm of **3.1**, and obtain $\mathcal{P}_{\mathrm{E}}(t + 1)$.
14. Send the elite solution individual in the training data to all the islands in the offline search module as the immigrant.
15. Select the best I environment individuals in $\mathcal{P}_{\mathrm{E}}(t+1)$, and set these environment individuals to islands of the offline search module as new identified environment $E_i(t+1)$. The pair of the environment individual and the island is determined by the similarity of the environment individual and the old identified environment $E_i(t)$ not to change the identified environment in each island drastically.
16. Let $t = t + 1$. If $t < N_{\mathrm{g}}$, go to Step 9, otherwise terminate the algorithm.

5 Dynamic Environment Problem

In this paper, we assume the following as the dynamic environments to be adapted.

1. The adaptive system cannot perceive the change of the environment directly.

2. An environment is not identified at the beginning of the search.

3. The number of evaluations is limited.

In such a situation, the hybrid method of the online and the offline adaptation is required.

5.1 Nk-Landscape Problem

We use an Nk-landscape problem[16, 17] as an example. The Nk-landscape problem is a function $f : \mathcal{B}^N \rightarrow \mathcal{R}$. k is the number of the locus that each locus is epistatically connecting. Each locus is given a [0,1] real number according to a $k + 1$ binary string, where $k + 1$ loci are a locus itself and connecting k loci. A real number is decided randomly in each binary string and its number is fixed. The function value of Nk-landscape is the mean value of each locus number.

In this paper, we set $N = 16$ and $k = 1$. Since there is a choice between the immediate neighbors of each locus and neighbors chosen at random for connecting, we use the immediate neighbors.

We use the two different Nk-landscape problems and exchange them in certain interval as the dynamic environment problem. We call this problem changing Nk-landscape problem.

5.2 Setup of GA

The genetic representation in the offline search module is a straightforward one that uses a 16-bit binary code. The fitness is the value of the objective function. The genetic representation in the online search module is a real number vector which is the weight values of the neural network.

In the offline search module, the uniform crossover with unity crossover rate and the locus-wise mutation with mutation rate 0.0625 are used. The number of islands is set to 10, and the population size of each island is set to 20. The temperature T of ITDGA is set to 1.

In the online search module, the UNDX with recommended parameters[19] and the mutation described in **3.1** with the locus-wise mutation rate 0.002 are used. The population size is set to 40, and the number of training data is set to 20. The exclusive radius R is set to 2.

The maximum number of generations is set to 400, and the problem is changed in the generation 200.

We have applied the elitism to both the online search module and the the offline search module.

6 Computer Simulation

The changing Nk-landscape problem is solved by EIGA.

Figure 1 shows the search process of environment population in the online search module with exclusive radius $R = 2$ and ITDGA temperature $T = 1$.

In the early stages of the search, the best fitness value of environment population is low because there are no information about the environment and few training data. However, the best fitness value is improved remarkably and becomes almost optimum fitness value 0. In generation 200, the best fitness value falls down because of environmental change. In that case, the fitness value also improved within 10 generation. These results show that the online search module of EIGA can identify the environment and adapt to the dynamic environment.

Figure 2 shows the search process of solution population in the offline search module in the same simulation of Figure 1 The trend of the search process of solution population is similar to that of environment population. In the early stages of the search, the best fitness values of solution population is low, and the best fitness value is improved. However, the improvement speed of solution population is slower than that of environment population. This is because the solution population can search only after the environment population searches well and finds the stable environment individual since the identified environment which determines the fitness value of solution individuals is replaced in every generation. After the environmental change at generation 200, the search population can find the optimum solution. These results show that the offline search module of EIGA can find the solution by utilizing the identified environment.

This simulation results show that the EIGA can adapt to the unidentified dynamic environment.

7 Conclusion

In this paper, the authors propose *the Environment Identifying GA*(EIGA), which can adapt the dynamic environments as identifying the environment. The

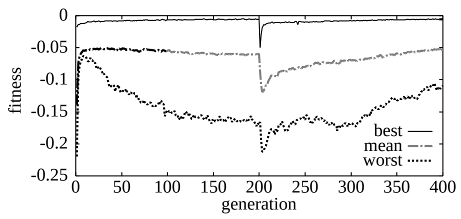

Figure 1: Search process of environment population in the online search module

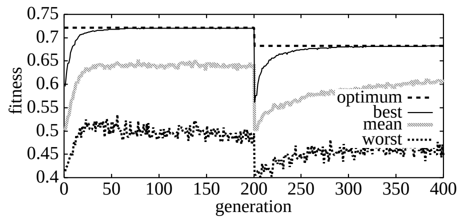

Figure 2: Search process of solution population in the offline search module

computer simulation is carried out to confirm the effectiveness of the EIGA, and a satisfactory performance is obtained by EIGA. Applying EIGA to more complicated problems is subjects of further study.

This research was supported by the Grant-in-Aid for Encouragement of Young Scientists(11750402) from the Japan Society for the Promotion of Science.

References

[1] J. H. Holland: *Adaptation in Natural and Artificial Systems*, The University of Michigan Press (1975)

[2] D. E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989)

[3] J. J. Grefenstette : Genetic algorithms for changing environments, *Problem Solving from Nature 2*, Vol. 2, pp.137-144 (1992)

[4] H. G. Cobb and J. J. Grefenstette : Genetic Algorithms for Tracking Changing Environments, *Proc. of 5th ICGA*, Vol. 5, pp. 523-529 (1993)

[5] N. Mori, J. Yoshida, H. Kita and Y. Nishikawa: A Thermodynamical Selection Rule for the Genetic Algorithm, *Proc. of 2nd IEEE Conference on Evolutionary Computation(ICEC'95)*, pp.188-192 (1995)

[6] N. Mori, H. Kita and Y. Nishikawa: Adaptation to a Changing Environment by Means of the Thermodynamical Genetic Algorithm, *Proc. of 4th PPSN* (*PPSN'96*), Vol. 4, pp. 513-522 (1996)

[7] N. Mori, H. Kita and Y. Nishikawa: Adaptation to a Changing Environment by Means of the Feedback Thermodynamical Genetic Algorithm, *Proc. of 5th PPSN*, Vol. 5, pp. 149-158 (1998)

[8] K. Ohkura and K. Ueda: A Genetic Algorithm for Nonstationary Function Optimization Problems, *Trans. SICE*, Vol. 8, No. 6, pp. 269-276 (1995)

[9] N. Mori, S. Imanishi, H. Kita, Y. Nishikawa: Adaptation to Changing Environments by Means of the Memory Based Thermodynamical Genetic Algorithm, *Proc. of 7th ICGA*, pp. 299-306 (1997)

[10] I. Kamihira, M. Yamaguchi and H. Kita: Online Adaptation of Vehicles by Means of an Evolutionary Control System, *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (*SMC '99*), pp. v-553-v558 (1999)

[11] F. Vavak, K. Jukes, and T.C. Fogarty: Adaptive balancing of bank of sugar-beet presses using a genetic algorithm with variable local search range, *3rd Intl. Mendel Conference on Genetic Algorithms*, pp. 164-169 (1997)

[12] T. Kude, N. Mori, K. Matsumoto: An Adaptive Evolutionary System by Mean of the Genetic Algorithm, *Proc. of SCI'2000*, pp. 541-542 (2000)

[13] T. Kude, N. Mori, K. Matsumoto: A Parallelization of the Island Thermodynamical Genetic Algorithm, *Proc. of SCI'99*, pp. 109-110 (1999)

[14] C. C. Petty and M. R. Leuze: A theoretical investigation of a parallel genetic algorithm, , *Proc. of 3rd Int. Conf. on Genetic Algorithms*, pp. 389-405 (1989)

[15] R. Tanese: Distributed genetic algorithms, *Proc. of 3rd Int. Conf. on Genetic Algorithms*, pp. 434-439 (1989)

[16] S. Kauffman: Adaptation on Rugged Fitness Landscapes, *Lectures in the Sciences of Complexity*, pp. 527-618 (1989)

[17] S. Kauffman: The origins of order, self-organization and selection in evolution, Oxford University Press (1993)

[18] D. E. Rumelhart et al.: *Parallel Distributed Processing*, MIT Press (1986)

[19] I. Ono and S. Kobayashi: A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, *Proc. of 7th ICGA*, pp. 246-253 (1997)

[20] Y. Ichii, N. Mori, K. Matsumoto: An Exclusive Radius Selection for the Real-coded Genetic Algorithm , *Proc. of SCI'2000*, pp. 119-120 (2000)