Application of self-organizing maps to genetic algorithms

S. Kan, Z. Fei & E. Kita
Graduate School of Information Sciences, Nagoya University, Japan

Abstract

This paper describes Self-Organizing Maps for Genetic Algorithm (SOM-GA). In this algorithm, the search performance of a real-coded genetic algorithm (RCGA) is enhanced with self-organizing map (SOM). The SOM is trained with the information of the individuals in the population. Sub-populations are generated from a whole population by the help of the map. The RCGA search is performed in the sub-populations. The Rastrigin function is considered as a test problem. The search performance of SOM-GA is compared with that of the RCGA. The results show that the use of the sub-population search algorithm improves the local search performance of the RCGA and therefore, SOM-GA can find better solutions in shorter CPU time than RCGA.
Keywords: real-coded genetic algorithms, self-organizing maps, Rastrigin function.

1 Introduction

Evolutionary Computations can be classified into Genetic Algorithms (GA)[1, 2], Genetic Programming (GP) [3, 4], Evolutionary Strategy (ES) [5] and so on. The GA, which is one of the most popular evolutionary computations, has been presented by Holland in 1970s. Since then, theoretical study of GA operators and the engineering application of GA have been studied widely by many researchers. Recently, their attention focuses on the multi-objective optimization and real-coded optimization problems from the viewpoint of engineering applications. Now, we will focus on the real-coded optimization problem.

An original genetic algorithm is designed for optimization problems with binary-coded design variables. Therefore, in the previous studies, real-coded design variables have to be transformed to binary-coded ones. However, some researchers

WIT Transactions on The Built Environment, Vol 106, © 2009 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/OP090011

have presented the real-coded GA (RCGA) in which the real-coded design variables are not transformed into the binary-coded ones [6–8]. In this study, we will present the RCGA with self-organizing maps (SOM), which is named as "Self-Organizing Maps for Genetic Algorithms (SOM-GA)".

The SOM, which has been presented by Kohonen [9] and Van Hulle [10], is a single layer feed-forward network where the output syntaxes are arranged in grid. Simply, the SOM-GA is the combinational algorithm of the Real-Coded Genetic Algorithm (RCGA) and the SOM clustering. The algorithm starts from the definition of the population of individuals. A self-organizing map is trained with the objective function and the design variables of the individuals in the population. The best match unit for a individual is chosen and then, a sub-population is defined by the individuals included in the circle centering on the best match unit. RCGA is performed iteratively in sub-populations and then, the obtained best individuals are added to the next whole population.

The combinational algorithm of the evolutionary algorithm and the SOM has already been presented by Buche *et al*. [11], which is named as Self-Organizing Maps for Multi-Objective Evolutionary Algorithms (SOM-MOEA). The SOM-MOEA is designed for the multi-objective optimization problems and therefore, their aim is different from the SOM-GA.

In the numerical example, Rastrign function is taken as a test function. The SOM-GA and RCGA are compared in their search performance by finding an optimal solution of the Rastrign function.

2 Self-organizing map

Before explanation of Self-Organizing Maps for Genetic Algorithms (SOM-GA), we will introduce Self-Organizing Maps [9, 10].

The self-organizing map is a single layer feed-forward network where the output syntaxes are arranged in grid (Fig. 1). Each input is connected to all output neurons. A weight vector with the same dimensionality as the input vectors is attached to every neuron. The number of input dimensions is usually a lot higher than the output grid dimension. SOMs are mainly used for dimensionality reduction rather than expansion.

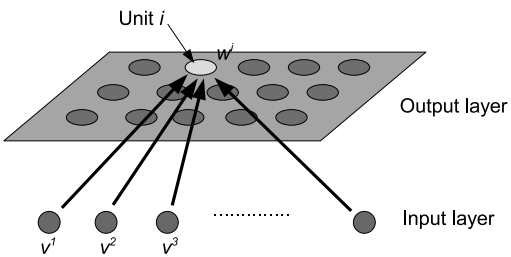

Figure 1: Self-organizing map.

The weight vector w^i at the unit i is given as

$$\boldsymbol{w}^i = \left\{w_1^i, w_2^i, \ldots, w_n^i\right\}^T, \tag{1}$$

and the input vector v^j is as

$$\boldsymbol{v}^j = \left\{v_1^j, v_2^j, \ldots, v_n^j\right\}^T. \tag{2}$$

Taking the Euclid distance $\left\|\boldsymbol{v}^j - \boldsymbol{w}^i\right\|$ as a norm, the best match unit is selected so that the norm is minimized. The best match unit $c\left(\boldsymbol{v}^j\right)$ can be defined as

$$c\left(\boldsymbol{v}^j\right) = \arg\min\{\left\|\boldsymbol{v}^j - \boldsymbol{w}^i\right\|\}. \tag{3}$$

Once determining $c(v^j)$, the weight vector is updated by

$$\boldsymbol{w}^i(t+1) = \boldsymbol{w}^i(t) + h_{ci}(t)\left\{\boldsymbol{v}^j(t) - \boldsymbol{w}^i(t)\right\}, \tag{4}$$

where t denotes the discretized time; $t = 0, 1, 2, \ldots$. The neighborhood function $h_{ci}(t)$ is defined as

$$h_{ci}(t) = \alpha_s \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right), \tag{5}$$

where r_i and r_c denote the position vectors of the unit i and the best match unit, respectively. The coefficient α_s is sometimes defined as a monotonically decreasing one. In this study, however, it is defined as the constant within the range of $0 < \alpha_s < 1$. The function $\sigma(t)$ is defined as

$$\sigma(t) = \sigma(t-1) - \frac{R}{TS}, \tag{6}$$

where $\sigma(0) = R$ and $\sigma(TS) = 0$. The parameters TS and R denote the number of training step and the initial radius, respectively.

The SOM algorithm is as follows.

1. Initialize randomly the weight vectors $\mathbf{w}^i$.
2. Select the unit $c(\mathbf{v}^j)$ so as to satisfy equation (3).
3. Update the weight vector $\mathbf{w}^i$ according to equation (4).
4. Update the neighborhood function according to equation (5).
5. Repeat from step 2 to step 4.

3 SOM-GA algorithm

For explanation of the SOM-GA algorithm, we will take the following notations.

$\boldsymbol{P}(t)$	Population at generation t.
$p_i(t)$	An individual in a population $\boldsymbol{P}(t)$.
M	Total number of individuals in a population $\boldsymbol{P}(t)$.
$\boldsymbol{SP}_i(t)$	Sub-populations generated from a population $\boldsymbol{P}(t)$.
Map	Self-Organizing Maps.
Rn	Radius for sub-population $\mathbf{SP_i(t)}$ in SOM-GA.

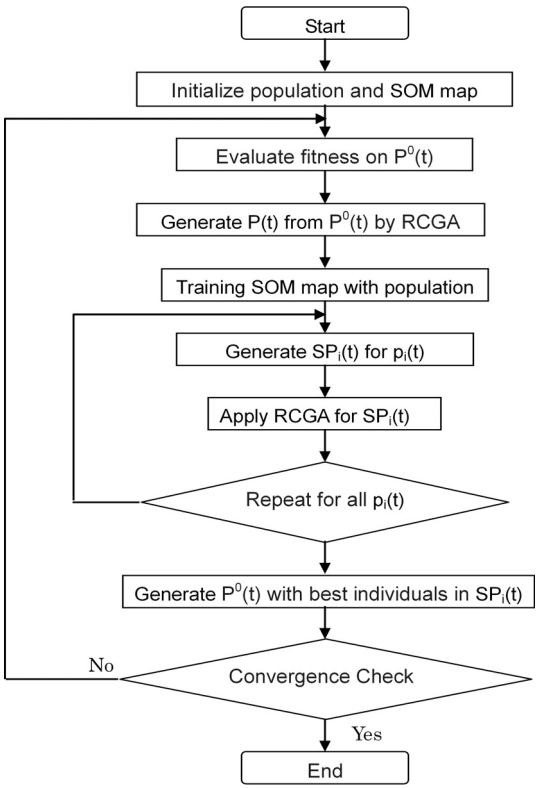

Figure 2: Flow chart of SOM-GA algorithm.

The SOM-GA algorithm (2) is as follows.

1. $t = 0$.
2. Initialize self-organizing map Map.
3. Generate randomly M individuals to construct an initial population $\boldsymbol{P}^0(t)$.
4. Evaluate fitness of individuals.
5. Generate the population $\boldsymbol{P}(t)$ as follows.
   (a) Select two individuals from the population $\boldsymbol{P}^0(t)$ by roulette selection.
   (b) Apply BLX-α crossover [6, 12] to generate new individuals.
   (c) Repeat until generating M individuals.
6. Train Map with the values of the objective function and the design variables of the individuals in the population $\boldsymbol{P}(t)$ according to the Section 3.
7. Generate new individuals as follows.
   (a) Define sub-populations on Map by all individuals within a radius of Rn from the best match unit for the individual $p_i(t)$.
   (b) Perform RCGA operation in the sub-population $\boldsymbol{SP}_i(t)$ to obtaining best individual $p_i(t+1)$.
   (c) Repeat for every individual $p_i(t)(i = 1, 2, \ldots, M)$.

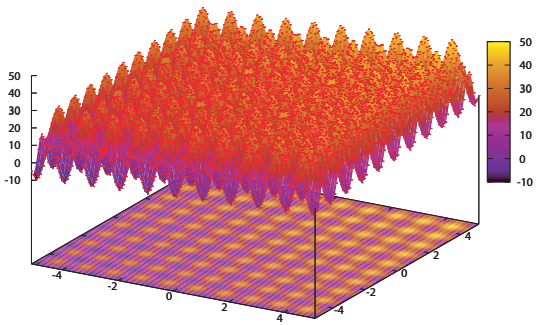

Figure 3: Rastrigin function in two-dimension.

8. Terminate process if convergence criterion is satisfied.
9. Construct new population $\boldsymbol{P}^0(t+1)$ by individuals $p_i(t+1)$ $(i = 1, 2, \ldots, M)$.
10. Update time step; $t \leftarrow t+1$.
11. Go to (4).

In the numerical examples, the test functions are taken as the objective functions.

4 Numerical examples

4.1 Problem setting

The SOM-GA is applied for finding an optimum solution of Rastrigin function. The test functions are taken as the objective functions and, the SOM map is trained with the values of the test functions and the design variables.

Rastrigin function is defined by

$$F(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$$
$$(-5.12 < x_i < 5.12) \tag{7}$$
$$\min(F(x)) = F(0, 0, \ldots, 0) = 0.$$

In case of two design variables, the function is shown in Fig. 3. Since Rastrigin function is a multi-modal function, it has one global minimum and many local minima. Design variables are independent each other.

In this paper, the number of the design variables is specified as $N_{dv} = 20$.

Common parameters for the SOM-GA and the RCGA are listed in Table 1. Number of the design variables is 20. Population size is 50. Crossover rate and the mutation rate are 1.0 and 0.002, respectively. Roulette selection and BLX-α crossover operations are adopted. In the real-coded mutation operation, the values of the design variables are changed randomly.

Table 1: Common parameters for both algorithms.

Number of design variables	$N_{dv} = 20$
Population size	$N_p = 50$
Selection	Roulette selection
Crossover	BLX-α, $\alpha = 0.5$
Crossover rate	1.0
Mutation rate	0.002
Number of trials	10

Table 2: Additional parameters for SOM-GA.

Map size and type	Hexagon of 20×20
Initial neighborhood radius	$R = 10$
Training rate	$\alpha_s = 0.8$
Training times	$TS = 1000$
Neighborhood radius	$Rn = 2$
Operation number in sub-population	$N_g^{sub} = 1000$

Table 3: Maximum generation step N_g.

	Rastrigin
$(N_g)_{RCGA}$	1,000,000
$(N_g)_{SOM-GA}$	20

Additional parameters for the SOM-GA are shown in Table 2. The neighborhood radius Rn is the radius of the circular region on the self-organizing map which covers a sub-population. The operation number in the subpopulation N_g^{sub} is the number of the RCGA operations in each sub-population.

Ten trials are performed in every cases from different initial populations. Mean values are shown in the numerical results.

Maximum generation steps N_g are shown in Fig. 3. In the SOM-GA, the RCGA operation is performed in every sub-populations. The estimation time of fitness function is equal to the product of the number of maximum generation step, the population size, and the number of the RCGA operations in every sub-populations. In order to equalize the estimation times of fitness functions in RCGA and

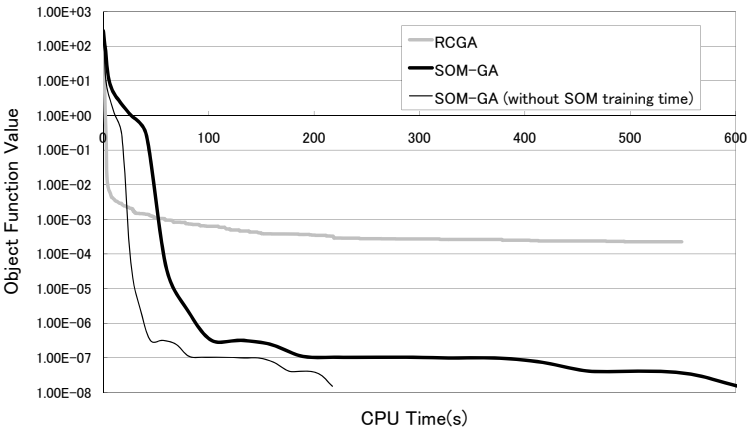

Figure 4: The comparison of RCGA and SOM-GA.

SOM-GA, the generation steps are specified according to

$$(N_g)_{RCGA} \geq (N_g)_{SOM-GA} \times N_p \times N_g^{sub}.$$

where $(N_g)_{RCGA}$ and $(N_g)_{SOM-GA}$ denote the maximum generation step in RCGA and SOM-GA, respectively.

4.2 Numerical results

Convergence histories of the best individuals are shown in Fig. 4. The abscissa and the ordinate denote the CPU time and the objective function of the best individual, respectively. The RCGA and SOM-GA results are labeled with "RCGA" and "SOM-GA", respectively. The convergence speed of RCGA is faster than that of the SOM-GA in the early steps. On the contrary, the convergence speed of SOM-GA is slower than the RCGA in the early steps and then, is accelerated when the objective function value is smaller than 0.001. A final solution is much better than that by RCGA.

The CPU time of the SOM-GA without SOM training time is also shown in the figure. The CPU time of the SOM-GA with SOM training time is twice as long as that without the SOM training time. This means that the computational cost for SOM training is almost equal to the remaining cost.

Next, we would like to discuss the effect of the map size to the search performance. The map size varies from 10×10 to 80×80. The results are compared in Fig. 5. The abscissa and the ordinate denote the dimensionless CPU time with the maximum CPU time of maximum map size and the value of objective function, respectively and the labels denote the map size. We notice that the CPU time increases according to the magnitude of the map size. When the map size is not small, i.e., greater than 30×30, the final solution converges to the same solution. Therefore, we can say in this case that the map size of 30×30 is best from the

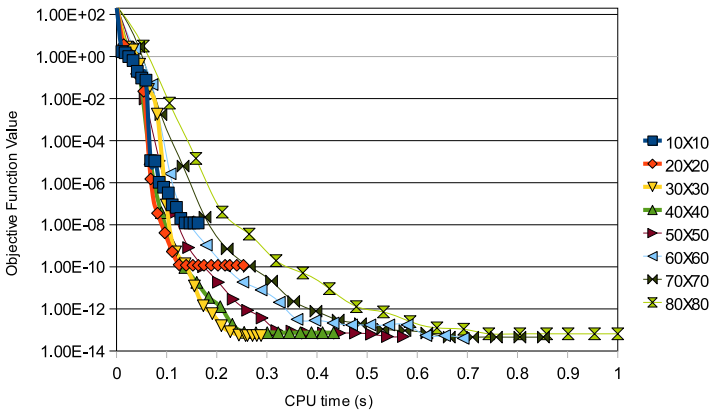

Figure 5: Effect of SOM map-size.

view-points of the search performance and the CPU time. Since the most adequate map size depends on the problem to be solved, we take a small map-size, i.e., the map size of 20×20, in the following examples.

5 Conclusions

This paper described the Self-Organizing Maps for Genetic Algorithms (SOM-GA) for real-valued single objective function problems. In the algorithm, the self-organizing maps are trained with the values of the objective function and the design variables of the individuals in a population and the sub-populations are defined by the help of the SOM clustering. The real-coded genetic algorithm (RCGA) is applied to the individuals in each sub-population. The processes are repeated until satisfying the convergence criterion.

The SOM-GA is applied for finding an optimum solution of Rastrigin function. The results show that the SOM-GA can find better solutions in shorter computational time than the original RCGA without SOM.

References

[1] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1 edition, 1975.
[2] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1 edition, 1989.
[3] J. R. Koza, editor. *Genwetic Programming II*. The MIT Press, 1994.
[4] J. R. Koza, F. H. Bennett III, D. Andre, and M. A. Keane, editors. *Genewtic Programming III*. Morgan Kaufmann Pub., 1999.

[5] T. Bäck, editor. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
[6] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval schemata. In L. D. Whitley, editor, *Foundation of Genetic Algorithms 2*, pp. 187–202. Morgan Kaufmann Publications, 1992.
[7] O. Takahashi, H. Kita, and S. Kobayashi. A real-coded genetic algorithm using distance dependent alternation model for complex function optimization. In Darrell Whitley, David Goldberg, Erick Cantu-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, editors, *Proc. of Genetic and Evolutionary Computation Conf. (GECCO2000)*, pp. 219–226, 2000.
[8] H. Kita. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms, evolutionary computation. *Evolutionary Computation*, Vol. 9, No. 2, pp. 223–242, 2001.
[9] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, 3 edition, 2001.
[10] M. M. Van Hulle. *Faithful Representations and Topographic Maps*. John Wiley & Sons, 2000.
[11] D. Buche, M. Milano, and P. Koumoutsakos. Self-organizing maps for multiobjective optimization. In A.M. Barry, editor, *GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference, New York, AAAI*, pp. 152–155, 2002.
[12] M. Takahashi and H. Kita. A crossover operator using independent component analysis forreal-coded genetic algorithms. In *Evolutionary Computation*, Vol. 1, pp. 643–649, 2001.