The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

A Real-Coded Genetic Algorithm Taking Account of the Weighted Mean of the Population

Naotoshi Nakashima[1], Yuichi Nagata[1], and Isao Ono[1]

[1]Tokyo Institute of Technology, Japan
(Tel: +81-45-924-5545, Fax: +81-45-924-5442)

[1]naoto@ic.dis.titech.ac.jp

Abstract: Continuous function optimization is an important problem in science and engineering. The real-coded genetic algorithm (RCGA) has shown good performance in continuous function optimization. AREX/JGG is one of the most promising RCGAs. However, we believe that AREX/JGG has two problems in terms of search efficiency. In this paper, we propose a new RCGA that overcomes the problems of AREX/JGG. In order to examine the effectiveness of the proposed RCGA, we compared the performance of the proposed RCGA with that of AREX/JGG on several benchmark problems in which initial populations do not cover the optimal points. As the result, we confirmed that the proposed RCGA succeeded in finding the optimal points faster than AREX/JGG.

Keywords: function optimization, genetic algorithms, real-coded GAs, AREX/JGG

1 Introduction

Continuous function optimization is an important problem that we often face in various domains such as science and engineering. In recent years, several real-coded Genetic Algorithms (RCGAs) for function optimization have been studied actively [1-4]. It has been reported that RCGAs are effective in optimizing high-dimensional functions with multimodality, epistasis and ill-scaleness. However, most conventional RCGAs have a serious problem that the performance deteriorates when the population does not cover the optimal point.

In order to remedy the above problem of the conventional RCGAs in the case where the population does not cover the optimum, Akimoto *et al*. has proposed AREX/JGG and reported that AREX/JGG shows better performance than the conventional RCGAs in benchmark problems in which initial regions do not cover the optimal points [3, 4]. However, we believe that AREX/JGG has the following two problems in terms of search efficiency: 1) If the population distribution does not cover the optimal point, AREX/JGG does not always generate offspring and eliminate individuals in the population so that the population distribution efficiently moves towards the optimal point. 2) If the population distribution covers the optimal point, AREX/JGG does not always generate offspring and eliminate individuals in the population so that the population efficiently converges towards the optimal point.

In this paper, we propose a new RCGA that overcomes the above two problems of AREX/JGG. In order to examine the effectiveness of the proposed method, we compare the performance of the proposed RCGA with that of AREX/JGG on several benchmark problems with multimodality, strong epistasis and ill-scaleness in which initial regions do not cover the optimal points.

2 AREX/JGG and Its Problems

2.1 Just Generation Gap (JGG)

JGG has been designed as a generation alternation model for multi-parent crossover operators [3, 4]. The algorithm of JGG is as follow.

(1) Initialization:
Randomly generate an initial population $\{p_k \mid k=1,\ldots,\tau\}$ within a specified region. Evaluate the object function values of all the individuals in the population. Initialize the generation number $\boldsymbol{g}$ with 1.

(2) Mating Selection: Randomly choose parents $\{\boldsymbol{y}_j|j=1,\ldots,n+1\}$ from the population $\{p_k|k=1,\ldots,\tau\}$, where n is the dimension of the problem.

(3) Generation of offspring: Generate offspring $\{\boldsymbol{x}_i|i=1,\ldots,\tau\}$ by applying AREX to parents $\{\boldsymbol{y}_j|j=1,\ldots,n+1\}$. Evaluate the object function values of the offspring $\{\boldsymbol{x}_i|i=1,\ldots,\tau\}$.

(4) Survival Selection: Select the best n+1 offspring among the offspring $\{\boldsymbol{x}_i|i=1,\ldots,\tau\}$ and remove the parents $\{\boldsymbol{y}_j|j=1,\ldots,n+1\}$ form the population $\{p_k|k=1,\ldots,\tau\}$.

(5) If termination conditions are satisfied, terminate the algorithm. Otherwise, $\boldsymbol{g} \leftarrow \boldsymbol{g}+1$ and go to step2.

2.2 AREX (Adaptive Real-coded Ensemble Crossover)

AREX has been proposed to remedy a serious problem of conventional real-coded GAs that premature convergence often occurs when the population does not cover the optimal point [3, 4].

AREX generates offspring by

$$\boldsymbol{x}_i = \boldsymbol{m} + \alpha \sum_{j=1}^{\mu} \epsilon_{i,j} \left(\boldsymbol{y}_j - \langle \boldsymbol{y} \rangle \right) \quad \cdots (1)$$

where $\boldsymbol{m}$ is the center of crossover and α is the expansion rate to be adapted.

The center of crossover $\boldsymbol{m}$ is calculated by

$$\boldsymbol{m} = \sum_{k=1}^{\mu} w_k \boldsymbol{y}_{k;\mu} \quad \cdots (2)$$

where $k;\mu$ denotes the best k individual out of μ parents. w_k (k=1,…,μ) denote weights where $\sum_{k=1}^{\mu} w_k = 1$ and are calculated by Eq.(3).

$$w_k = 2(\mu + 1 - k) / (\mu(\mu + 1)) \quad \cdots (3)$$

Using Eq.(2) as the center of crossover, AREX aims at prompting the center of mass of the population to move towards the promising region.

The expansion rate α is adapted so that offspring are generated wider than their parents in order to keep the diversity of the population when the population moves. The expansion rate α is adapted as follows:

$$\alpha^{(g+1)} := \max \left(\alpha^{(g)} \sqrt{(1 - c_\alpha) + c_\alpha \frac{L_{\mathrm{cdp}}}{L_{\mathrm{avg}}}}, 1 \right) \quad \cdots (4)$$

where the g and c_α are the generation number and the learning rate, respectively. L_{cdp} is the Mahalanobis distance between the center of crossover $\boldsymbol{m}$ and the mean of the best μ offspring $\langle x \rangle_\mu$. L_{cdp} is defined by Eq.(5) and is calculated by Eq.(6).

$$L_{\mathrm{cdp}} = (\langle \boldsymbol{x} \rangle_\mu - \boldsymbol{m})^{\boldsymbol{T}} \boldsymbol{C}^{+} (\langle \boldsymbol{x} \rangle_\mu - \boldsymbol{m}) \quad \cdots (5)$$

$$= \alpha^2 (\mu - 1) \left(\sum_{j=1}^{\mu} \langle \epsilon_j \rangle_\mu^2 - \frac{1}{\mu} \left(\sum_{j=1}^{\mu} \langle \epsilon_j \rangle_\mu \right)^2 \right) \quad \cdots (6)$$

C is the covariance matrix of the offspring distribution and C^+ is the Moore-Penrose inverse matrix of C. The $i{:}\lambda$ denotes the index of the best i-th individual among the λ offspring. L_{avg} denotes the expected value of the squared distance under random selection and is calculated by Eq.(7).

$$L_{\mathrm{avg}} = \alpha^2 \sigma^2 (\mu - 1)^2 / \mu \quad \cdots (7)$$

2.3 Problems of AREX/JGG

AREX/JGG generates an offspring distribution near around n+1 parents randomly chosen from the population and replaces the n+1 parent with the n+1 best offspring. Therefore, assuming big-valley functions, we believe that AREX/JGG has two problems in terms of search efficiency.

[Problem 1] As shown Fig.1, if the population distribution does not cover the optimal point in the search space, AREX/JGG does not always generate offspring in the outer area of the population distribution near the optimal point, eliminate individuals in the outer area of the population distribution far from the optimal point and extend the population distribution. Therefore, we believe that AREX/JGG cannot effectively move the population distribution toward the optimal point.

[Problem 2] As shown Fig.2, if the population distribution covers the optimal point, AREX/JGG does not always generate offspring near around the optimal point and eliminate individuals in the outer area of the population distribution. Furthermore, if the offspring distribution is not generated so as to cover the optimal point, the expansion rate in Eq.(4) increases, which means that the search becomes inefficient. Therefore, we believe that AREX/JGG cannot efficiently shrink the population distribution to the optimal point.

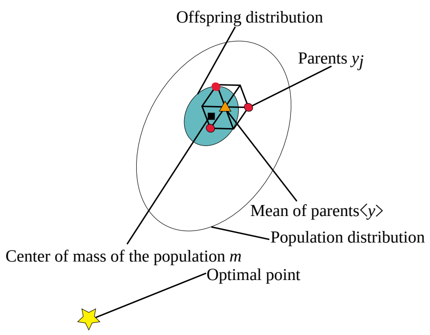

Fig.1. Offspring distribution generated by AREX/JGG when the population does not cover the optimal point.

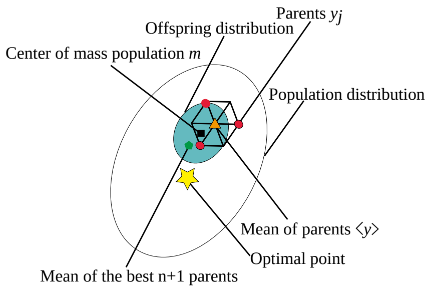

Fig.2. Offspring distribution generated by AREX/JGG when the population covers the optimal point.

3 The Proposed RCGA

3.1 Basic ideas

In this section, we describe our basic ideas to overcome the problems of AREX/JGG pointed out in section 2.3. If the population does not cover the optimal point, we generate offspring in the outer area of the population near the optimal point and eliminate individuals far from the optimal point from the population. This is expected to result in the population quickly moving towards the optimal point. On the other hand, if the population distribution covers the optimal point, we generate offspring near around the optimal point and eliminate individuals in the outer area of the population. By doing this, we expect that the population distribution converges to the optimal point quickly.

Based on the above basic ideas, we propose a new RCGA that consists of the following mating selection method, crossover operator and survival selection method.

Mating selection: The proposed RCGA chooses the n+1 worst individuals in the population as parents. As the result, as shown in Fig.3, if the population distribution does not cover the optimal point, the proposed mating selection method is expected to choose individuals in the outer area of the population distribution far from the optimal point. On the other hand, if the population distribution covers the optimal point, as shown in Fig.4, the proposed mating selection method is expected to choose individuals in the outer area of the population distribution.

Crossover: As shown in Fig.3 and Fig.4, the proposed RCGA generates an offspring distribution by crossover whose center is a weighted mean of the all individuals in the population. As the result, as shown in Fig. 3, if the population distribution does not cover the optimal point, the proposed RCGA is expected to generate offspring also in the outer area of the population near the optimal point. On the other hands, if the population distribution covers the optimal point, as shown in Fig.4, the proposed RCGA is expected to generate many offspring near around the optimal point. Furthermore, the proposed RCGA also adapts the expansion rate according to the Mahalanobis distance between the center of crossover and the mean of the best n+1 offspring as AREX does. While the expansion rate in AREX/JGG is only for expanding the population distribution in order to avoid the premature convergence, the expansion rate in the proposed RCGA is for not only expanding the population distribution but also shrinking it. As the result, if the population distribution covers the optimal point, the proposed RCGA is expected to be able to shrink the population distribution in the optimal point quickly.

Survival selection: The proposed RCGA replaces the n+1 worst parents in the population with the n+1 best offspring. As the result, if the population distribution does not cover the optimal point, as shown in the Fig.3, the proposed RCGA is expected to eliminate individuals far from the optimal point in the population distribution. On the other hand, if the population distribution covers the optimal point, as shown in the Fig.4, the proposed RCGA is expected to remove the individuals in the outer area of the population distribution.

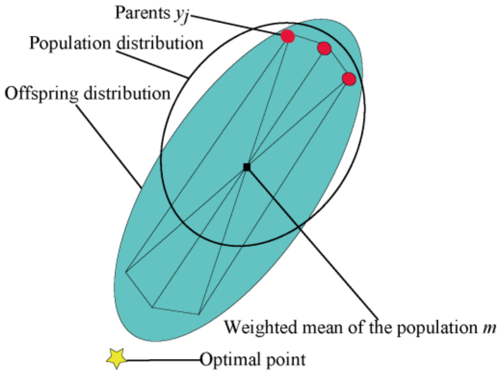

Fig.3. Offspring distribution generated by the proposed RCGA when the population does not cover the optimal point.

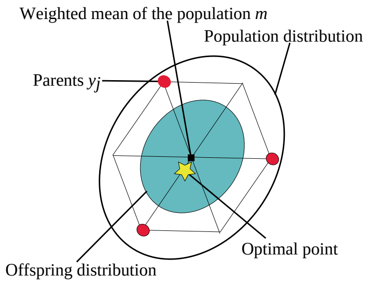

Fig.4. Offspring distribution generated by the proposed RCGA when the population covers the optimal point.

3.2 Algorithm

The algorithm of the proposed RCGA is as follows.

(1) Initialization: Generate an initial population $\{p_k|k=1,\ldots,\tau\}$ randomly within a specified region. Evaluate the objective function values of all the individuals in the population. Initialize the generation number $\boldsymbol{g}$ with 1.

(2) Mating Selection: Choose the n+1 worst individuals in the population $\{p_k|k=1,\ldots,\tau\}$ as parents $\{\boldsymbol{y}_j|j=1,\ldots,n+1\}$, where n is the dimension of the problem.

(3) Generation of offspring: Generate offspring $\{\boldsymbol{x}_i|i=1,\ldots,\tau\}$ with the parents $\{\boldsymbol{y}_j|j=1,\ldots,n+1\}$ according to Eq.(8) and evaluate the objective function values of the offspring $\{\boldsymbol{x}_i|i=1,\ldots,\tau\}$.

$$\boldsymbol{x}_i = \boldsymbol{m}' + \alpha \sum_{j=1}^{\mu} \epsilon_{i,j} \left(\boldsymbol{y}_j - \boldsymbol{m}'\right) \quad \cdots (8)$$

$\boldsymbol{m}'$ is a weighted mean vector calculated by Eq.(9).

$$\boldsymbol{m}' = \sum_{k=1}^{\tau_\alpha} w_k \boldsymbol{p}_{k;\tau} \quad \cdots (9)$$

$k;\tau$ denotes the index of the best k individual in the population whose size is τ. $\boldsymbol{w}_k$ is a linear weight where $\sum_{k=1}^{\mu} w_k = 1$. $\boldsymbol{w}_k$ is calculated by using the best τ_α individuals in the population $\{p_k|k=1,\ldots,\tau\}$ according to Eq.(10).

$$w_k = 2\left(\tau_\alpha + 1 - k\right) / \left(\tau_\alpha\left(\tau_\alpha + 1\right)\right) \quad \cdots (10)$$

α in Eq.(8) is the expansion rate and is given by Eq.(11).

$$\alpha^{(g+1)} = \alpha^{(g)} \sqrt{(1 - c_\alpha) + c_\alpha \frac{L_{\text{cdp}}}{L_{\text{avg}}}} \quad \cdots (11)$$

$$L_{\text{cdp}} = \alpha^2(\mu-1)\left(\sum_{j=1}^{\mu} \langle\epsilon_j\rangle_\mu^2 - \frac{1}{\mu}\left(\sum_{j=1}^{\mu}\langle\epsilon_j\rangle_\mu\right)^2\right) \quad \cdots (12)$$

$$L_{\text{avg}} = \alpha^2\sigma^2(\mu-1)^2/\mu \quad \cdots (13)$$

where $\langle\varepsilon_j\rangle_\mu = \sum_{i=1}^{\mu} \varepsilon_{i:\lambda,j}/\mu$ and $\varepsilon \sim \boldsymbol{N}(0, \sigma^2)$. $i{:}\lambda$ is the index of the best i individual of the λ offspring.
Note that the proposed RCGA allows the expansion rate to become smaller than 1.0 while that in AREX/JGG is always larger than or equal to 1.0.

(4) Survival Selection: Sort the offspring $\{\boldsymbol{x}_i|i=1,\ldots,\tau\}$ by the objective function values. Replace the parents

{y_j|j=1,…,n+1} in the population with the n+1 best offspring.

(5) Termination Condition: If termination conditions are satisfied, terminate the algorithm. Otherwise, $\boldsymbol{g} \leftarrow \boldsymbol{g}$+1 and go to step 2.

3 Experiments

In order to examine the effectiveness of the proposed RCGA, we compared the performance of the proposed RCGA with that of AREX/JGG. The system parameters of AREX/JGG are set to the recommended values in the literature [4].

We use the 20-dimensional unimodal benchmark functions (Sphere, Rosenbrock-Star, Rosenbrock-Chain, Ellipsoid, k-Tablet) and the 20-dimensional multimodel benchmark functions (Ackley, Bohachevsky, Schaffer, Rastrigin) as shown in Table 1. The optimal points of the Rosenbrock-Star function and the Rosenbrock-Chain function are $\boldsymbol{x}_{\mathrm{opt}}=[1,\dots,1]^T$ and the optimal points of the other functions are $\boldsymbol{x}_{\mathrm{opt}}=[0,\dots,0]^T$. The function value of the optimal point of each function $f(x_{opt})$ is zero. The initial regions are set to $[1,5]^{20}$ for the Sphere, Ellipsoid, k-tablet and Rastrigin functions, $[-2,2]^{20}$ for the Rosenbrock-Star and Rosenbrock-Chain functions, $[1,30]^{20}$ for the Ackley function, $[1,15]^{20}$ for the Bohachevsky function and $[1,100]^{20}$ for the Schaffer function.

We made preliminary experiments to determine the population size and the number of offspring per crossover for the combination of each algorithm and each benchmark problem. They were determined so that the average number of evaluations becomes the smallest under the constraint that the algorithm succeeded in finding the optimum on the benchmark problem in all the ten trials. We use the average number of evaluations as a performance index.

Table 2 shows the experimental result. The proposed RCGA succeeded in finding the optimal points faster than AREX/JGG in all the benchmark functions. This result suggests that the proposed RCGA is more effective than AREX/JGG.

Table 1. : Benchmark functions.

Function	Definition
f_1 : Sphere	$f_1 = \sum_{i=1}^{n} x_i^2$
f_2 : Elliposoid	$f_2 = \sum_{i=1}^{n} (1000^{i-1/n-1} x_i)^2$
f_3 : K_tablet	$f_3 = \sum_{i=1}^{n} x_i^2 + \sum_{i=k+1}^{n} (100x_i)^2$
f_4 : Rosenbrock_Star	$f_4 = \sum_{i=1}^{n} 100(x_i^2 - x_1)^2 + (x_i - 1)^2$
f_5 : Rosenbrock_Chain	$f_5 = \sum_{i=1}^{n} 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2$
f_6 : Ackley	$f_6 = 20 - 20exp(-0.2\sqrt{1/n \sum_{i=1}^{n} x_i^2})$ $+e - exp(1/n \sum_{i=1}^{n} \cos(2\pi x_i))$
f_7 : Bohachevsky	$f_7 = \sum_{i=1}^{n-1} (x_i^2 + 2x_{i+1}^2$ $-0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$
f_8 : Schaffer	$f_8 = \sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25}$ $\times(\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1.0)]$
f_9 : Rastrigin	$f_9 = 10n + \sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i)$

Table 2. : The average numbers of evaluations of AREX/JGG and the proposed RCGA. The numbers in the parentheses are the population size and the number of offspring per crossover, respectively, where n (=20) is the dimension of the problem. Ratio is the number of evaluations of the proposed RCGA divided by that of AREX/JGG multiplied by 100. If the ratio is smaller than 100%, the proposed RCGA succeeded in finding the optimum with fewer evaluations than AREX/JGG.

f	AREX/JGG	Proposed RCGA	Ratio
f_1	2.25e4 ($5n$, $4n$)	**1.34e4 (5n, 3n)**	60%
f_2	3.39e4 ($6n$, $3n$)	**1.68e4 (6n, 3n)**	50%
f_3	5.12e4 ($6n$, $3n$)	**2.72e4 (6n, 4n)**	53%
f_4	5.96e4 ($9n$, $3n$)	**3.21e4 (6n, 3n)**	54%
f_5	1.06e5 ($5n$, $4n$)	**6.71e4 (6n, 4n)**	63%
f_6	4.21e4 ($6n$, $3n$)	**2.48e4 (6n, 4n)**	59%
f_7	4.32e4 ($10n$, $4n$)	**1.76e4 (8n, 3n)**	41%
f_8	2.08e5 ($16n$, $3n$)	**9.42e4 (14n, 6n)**	45%
f_9	1.83e5 ($25n$, $4n$)	**1.49e5 (50n, 8n)**	81%

4. Conclusions

In this paper, we proposed a new RCGA in order to overcome the problems of AREX/JGG. The proposed RCGA consists of the mating selection that chooses the n+1 worst individuals in the population as parents, the crossover that employs a weighted mean of the population as the center of the crossover and the survival selection that replaces the n+1 worst parents in the population with the n+1 best offspring. We confirmed that the proposed RCGA outperformed AREX/JGG on various benchmark functions through numerical experiments.

We have a plan to evaluate the performance of the proposed RCGA in higher-dimensional problems and, if necessary, we have to improve the proposed RCGA. We also will apply the proposed RCGA to difficult real-world applications such as the lens design problem in order to investigate the effectiveness of the proposed RCGA.

References

[1] Herrera, F., Lozano, M. and Sanchez, A.M. : A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study, International Journal of Intelligent Systems, 18:309–338 (2003).

[2] Sinha, A., Srinivasan, A. and Deb, K. : A population-based, parent centric procedure for constrained real-parameter optimization. Proc. 2006 IEEE Congress on Evolutionary Computation (CEC2006), pp. 239–245 (2006).

[3] Akimoto, Y., Sakuma, J., Ono, I. and Kobayashi, S.: Adaptation of expansion rate for real-coded crossovers, Proc. Genetic and Evolutionary Computation Conference 2009 (GECCO2009), pp.739-746 (2009).

[4] Akimoto, Y., Nagata, Y., Sakuma, J., Ono, I. and Kobayashi, S. : Proposal and evaluation of adaptive real-coded crossover AREX, Transactions of the Japanese Society for Artificial Intelligence, Vol. 24, No. 6, pp. 446-458 (2009) (in Japanese).