

# Revision of a Floating-Point Genetic Algorithm GENOCOP V for Nonlinear Programming Problems

K. Kato\*, M. Sakawa and H. Katagiri

*Department of Artificial Complex Systems Engineering, Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima, Hiroshima, Japan*

**Abstract:** In this paper, focusing on general nonlinear programming problems, we attempt to propose a general and high-performance approximate solution method for them. In recent years, S. Koziel *et al.* have proposed a floating-point genetic algorithm, GENOCOP V, as a general approximate solution method for nonlinear programming problems and showed its efficiency, there are left some shortcomings of the method. In this paper, incorporating ideas to cope with these shortcomings, we propose a revised GENOCOP V (RGENOCOP V). Furthermore, we show the efficiency of the proposed method RGENOCOP V by comparing it with two existing methods, RGENOCOP III and GENOCOP V through the application of them into the numerical examples.

## INTRODUCTION

A nonlinear programming problem is called a convex programming problem when its objective function and its constraint region are convex. For such convex programming problems, there have been proposed many efficient solution methods as the successive quadratic programming method [1], the generalized reduced gradient method [2] and so forth. Unfortunately, no decisive solution method has been proposed for nonconvex programming problems. For such nonconvex programming problems, as practical solution methods, genetic algorithms have been widely used.

Genetic algorithms proposed by Holland [3] have attracted considerable attention as global methods for complex function optimization since De Jong considered genetic algorithms in a function optimization setting [4]. However, many of the test function minimization problems solved by a number of researchers during the past 20 years involve only specified domains of variables. Only recently several approaches have been proposed for solving general nonlinear programming problems through genetic algorithms [4-6].

In 1995, Michalewicz *et al.* [7] proposed a floating-point genetic algorithm GENOCOP III and showed its efficiency for nonconvex programming problems. In GENOCOP III, the individual representation using the floating-point format is adopted. Furthermore, in GENOCOP III, two populations are used: one is the search population consisting of individuals which satisfies linear constraints, the other is the reference population consisting of individuals which satisfies all constraints. However, it is known that there exist drawbacks about the generation of initial individuals and the generation of offsprings in the crossover operation. In recent years, Sakawa *et al.* [8] proposed the revised,

GENOCOP III (RGENOCOP III) involving improvements to deal with these drawbacks of GENOCOP III. On the other hand Koziel *et al.* [9] proposed GENOCOP V which is based on the homomorphous mapping as a new floating-point genetic algorithm and showed its superiority to GENOCOP III in processing time.

Unfortunately, some shortcomings exist in GENOCOP V. When the homomorphous mapping is used in GENOCOP V, we need to find at least one feasible solution which is called the basepoint solution.

Since GENOCOP V adopts the method to generate the fixed number of solutions in order to find the basepoint solution, it often happens that GENOCOP V cannot start the search because no feasible solution is found even if the problem is feasible. In addition, the precision of (approximate) optimal solutions obtained by GENOCOP V is not good since they greatly depend on the basepoint solution of the homomorphous mapping. Since only one basepoint solution is used in the whole search process in GENOCOP V, the precision of obtained approximate optimal solutions may not be good.

In this research, we propose a revised version of GENOCOP V (RGENOCOP V) by considering a new way to find the basepoint solution where a problem to minimize the amount of constraint violation is solved as well as a new method to select and update the basepoint solution.

## NONLINEAR PROGRAMMING PROBLEMS

In this research, we consider general nonlinear programming problem with constraints formulated as:

$$\begin{aligned} & \text{minimize } f(\mathbf{x}) \\ & \text{subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & \quad \quad \quad l_j \leq x_j \leq u_j, \quad j = 1, 2, \dots, n \\ & \quad \quad \quad \mathbf{x} = (x_1, x_2, \dots, x_n)^T \in R^n \end{aligned} \quad (1)$$

where  $f$  and  $g_i$  are convex or nonconvex real-valued functions,  $l_j$  and  $u_j$  are the lower bound and the upper bound of each decision variable  $x_j$ . In the following, we denote the feasible region of (1) by  $X$ .

\*Address correspondence to this author at the Department of Artificial Complex Systems Engineering, Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima, Hiroshima, Japan; E-mail: kosuke-kato@hiroshima-u.ac.jp

**GENOCOP V**

In 1999, Koziel *et al.* [9] proposed GENOCOP V, a floating-point genetic algorithm incorporating the homomorphous mapping.

**Homomorphous Mapping**

In [9], they considered a mapping  $T: F \rightarrow [-1,1]^n$ , shown in Fig. (1), where  $X$  is the feasible region of the problem and  $[-1,1]^n$  is an  $n$  dimensional hypercube.

- (1) The mapping  $T$  maps some point  $r \in X$  to the origin  $0 \in [-1,1]^n$ , i.e.,  $T: r \mapsto 0$ . The solution  $r$  is called the basepoint solution of  $T$ .
- (2) The mapping  $T$  maps each point  $x \in X$  to a point  $y \in [-1,1]^n$ , determined as:

$$T: x \mapsto y = \frac{x - r}{t_{\max} \cdot \max \{x_1 - r_1, x_2 - r_2, \dots, x_n - r_n\}} \quad (2)$$

where  $t_{\max}$  is a positive real number such that  $r + (x - r) \cdot t_{\max}$  is on the boundary of  $X$ .

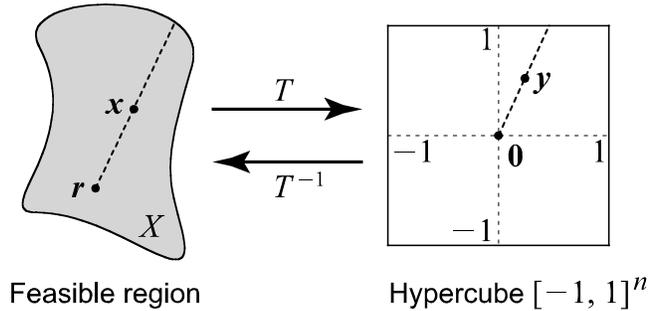


Fig. (1). Homomorphous mapping  $T$ .

The mapping  $T$  can be executed fast and the local relation of points is preserved.. In particular, if we use the  $n$  dimensional hypercube  $[-1,1]^n$  as the search space in place of the feasible region  $X$ , we do not have to prepare special genetic operators to keep the feasibility of solutions since  $[-1,1]^n$  is convex. Thereby, the improvement of the search efficiency of the genetic algorithm is expected by the introduction of the homomorphous mapping.

**Computational Procedures of GENOCOP V**

Step 1: Calculate the upper bound and the lower bound for each decision variable  $x_j$  based on linear constraints and upper-lower bound constraints, and generate individuals satisfying these constraints at random. If there is a solution to satisfy all constraints of the problem, i.e., there is a feasible solution, it becomes the basepoint solution  $r$ . If not, repeat the generation of individuals satisfying the above constraints at random several times. Go to step 2. If we can find no feasible solution through all trials, it is judged that the problem is infeasible and the procedure is terminated.

Step 2: Generate initial individuals in the  $n$  dimensional hypercube  $[-1,1]^n$  randomly. Go to step 3.

Step 3: Calculate a solution in the feasible region  $X$  corresponding to each individual in the  $n$  dimensional

hypercube  $[-1,1]^n$  by using the inverse mapping  $T^{-1}$  of the homomorphous mapping  $T$  based on the basepoint solution  $r$ , and evaluate all solutions in  $X$ . Go to step 4.

Step 4: If the termination conditions are satisfied, the procedure is terminated. Otherwise, go to step 5.

Step 5: Apply reproduction operator. Go to step 6.

Step 6: Select an operator randomly from among seven operators: uniform mutation, boundary mutation, non-uniform mutation, whole arithmetical crossover, simple arithmetical crossover, whole non-uniform mutation, a version of whole arithmetical crossover (some of them are shown in Fig. (2)), and apply the selected operator. Go to step 3.

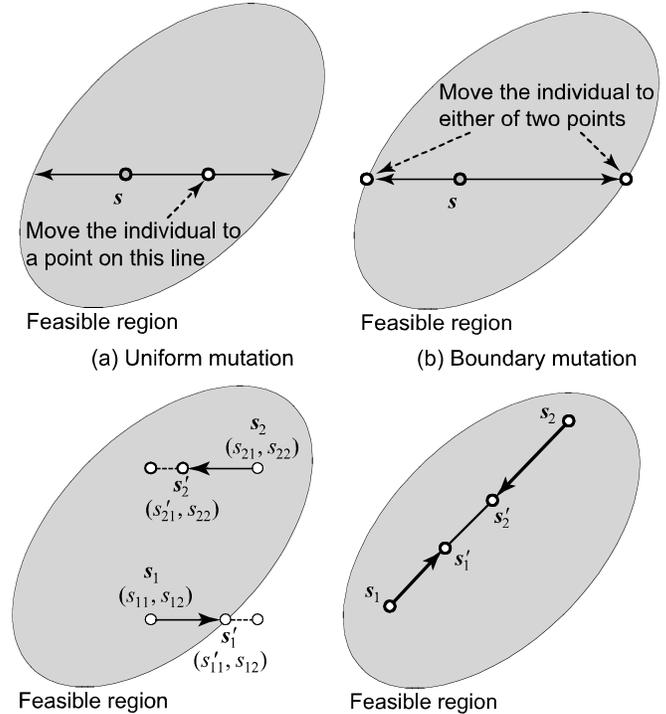


Fig. (2). Typical operators in step 6.

However, there exist the following shortcomings in GENOCOP V.

- (1) Since GENOCOP V adopts the method to generate the fixed number of solutions in order to find the basepoint solution, it often happens that GENOCOP V cannot start the search because no feasible solution is found even if the problem is feasible. In addition, the precision of (approximate) optimal solutions obtained by GENOCOP V is not good since they greatly depend on the basepoint solution of the homomorphous mapping.
- (2) It is known that approximate optimal solutions obtained by GENOCOP V greatly depend on the basepoint solution of the homomorphous mapping. Since only one basepoint solution is used in the whole search process in GENOCOP V, the precision of obtained approximate optimal solutions may not be good.

**REVISION OF GENOCOP V**

In this section, we propose the revised GENOCOP V which resolves shortcomings of GENOCOP V mentioned in the previous section.

**Generation of Initial Basepoint Solution**

In GENOCOP V [9], we have to obtain at least one feasible solution before the first mapping. However, we often cannot find any feasible solutions as initial feasible solutions since they are searched by randomly generating a given number of solutions which satisfy both upper bound constraints and lower ones. Thereby, the shortcoming (1) will be resolved by using the following optimization problem to minimize the degree of violation of constraints like RGENOCOP III [8].

$$\begin{aligned} &\text{minimize } \sum_{i=1}^m \{g_i(\mathbf{x})\}^2 \\ &\text{subject to } l_j \leq x_j \leq u_j, j = 1, 2, \dots, n \\ &\quad \mathbf{x} = (x_1, x_2, \dots, x_n)^T \in R^n \end{aligned} \tag{3}$$

**Updating of Basepoint Solutions**

In RGENOCOP V proposed in this paper, the homomorphous mapping is used as in GENOCOP V. As pointed out in the shortcoming (2) of GENOCOP V, the precision of obtained approximate optimal solutions greatly depends on the basepoint solution of the homomorphous mapping. In order to improve the precision, updating basepoint solutions, i.e., substituting the best individual in the current solution for the current basepoint solution  $r$  is adopted in RGENOCOP V, as shown in Fig. (3).

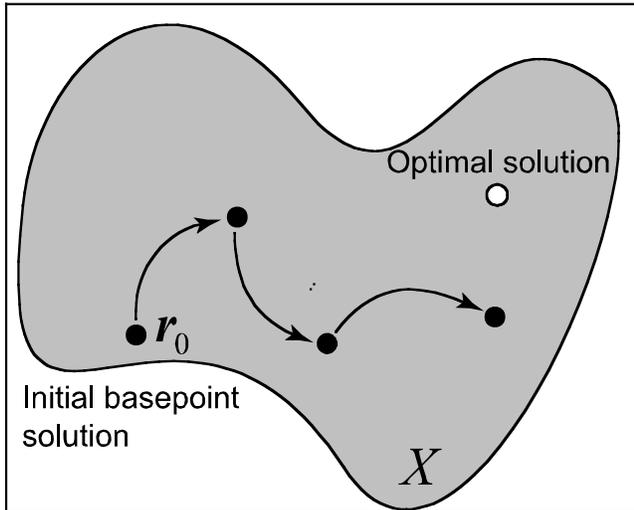


Fig. (3). Update of the basepoint  $r$  of the homomorphous mapping.

**NUMERICAL EXPERIMENTS**

In this section, we show the efficiency of the proposed RGENOCOP V through numerical experiments. We use five test problems in [9], and apply RGENOCOP III [8], GENOCOP V [9] and RGENOCOP V to them.

**Problem #1**

$$\begin{aligned} &\text{minimize } f_1(\mathbf{x}) = \sum_{j=1}^4 5x_j - \sum_{j=1}^4 5x_j^2 - \sum_{j=5}^{13} x_j \\ &\text{subject to } 2x_1 + 2x_2 + x_{10} + x_{11} \leq 10 \\ &\quad 2x_1 + 2x_3 + x_{10} + x_{12} \leq 10 \\ &\quad 2x_2 + 2x_3 + x_{11} + x_{12} \leq 10 \\ &\quad -8x_1 + x_{10} \leq 0, \quad -8x_2 + x_{11} \leq 0 \\ &\quad -8x_3 + x_{12} \leq 0, \quad -2x_4 - x_5 + x_{10} \leq 0 \\ &\quad -2x_6 - x_7 + x_{11} \leq 0, \quad -2x_8 - x_9 + x_{12} \leq 0 \\ &\quad 0 \leq x_j \leq 1, j = 1, 2, \dots, 9, \quad 0 \leq x_{13} \leq 1 \\ &\quad 0 \leq x_j \leq 100, j = 10, 11, 12 \end{aligned}$$

The objective function  $f_1$  is quadratic and its globally optimal solution is known as follows.

$$\mathbf{x}^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1), \quad f_1(\mathbf{x}^*) = -15.$$

For the optimal solution, six constraints are active.

**Problem #2**

$$\begin{aligned} &\text{minimize } f_2(\mathbf{x}) = x_1 + x_2 + x_3 \\ &\text{subject to } 0.0025(x_4 + x_6) - 1 \leq 0 \\ &\quad 0.0025(x_5 + x_7 - x_4) - 1 \leq 0 \\ &\quad 0.01(x_8 - x_5) - 1 \leq 0 \\ &\quad -x_1x_6 + 833.33252x_4 + 100x_1 \\ &\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad -83333.331 \leq 0 \\ &\quad -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\ &\quad -x_3x_8 + 125000 + x_3x_5 - 2500x_5 \leq 0 \\ &\quad 100 \leq x_1 \leq 10000, \quad 1000 \leq x_2 \leq 10000 \\ &\quad 1000 \leq x_3 \leq 10000 \\ &\quad 10 \leq x_j \leq 1000, j = 4, 5, \dots, 8 \end{aligned}$$

The objective function  $f_2$  is linear and its globally optimal solution is as follows.

$$\begin{aligned} \mathbf{x}^* &= (579.3167, 1359.943, 5110.071, 182.0174, \\ &\quad 295.5985, 217.9799, 286.4162, 395.5979), \\ f_2(\mathbf{x}^*) &= 7049.330923. \end{aligned}$$

For the optimal solution, all constraints are active.

**Problem #3**

$$\begin{aligned} &\text{minimize } f_3(\mathbf{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 \\ &\quad + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 \\ &\quad + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\ &\text{subject to } -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 \\ &\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad + 5x_5 \leq 0 \\ &\quad -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \end{aligned}$$

$$\begin{aligned}
 & - 196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\
 & 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 \\
 & \quad + 5x_6 - 11x_7 \leq 0 \\
 & - 10 \leq x_j \leq 10, \quad j = 1, 2, \dots, 7
 \end{aligned}$$

The objective function  $f_3$  is polynomial and its globally optimal solution is as follows.

$$\begin{aligned}
 \mathbf{x}^* &= (2.330499, 1.951372, -0.4775414, \\
 & 4.365726, -0.6244870, 1.038131, 1.594227), \\
 f_3(\mathbf{x}^*) &= 680.6300573.
 \end{aligned}$$

For the optimal solution, two constraints are active.

**Problem #4**

$$\begin{aligned}
 \text{minimize } f_4(\mathbf{x}) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 \\
 & + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\
 & + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 \\
 & + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \\
 \text{subject to } & - 105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\
 & 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\
 & - 8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\
 & 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 \\
 & \quad - 7x_4 - 120 \leq 0 \\
 & 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\
 & x_1^2 + (x_2 - 2)^2 - 2x_1x_2 \\
 & \quad + 14x_5 - 6x_6 \leq 0 \\
 & 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 \\
 & \quad - x_6 - 30 \leq 0 \\
 & - 3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \\
 & - 10 \leq x_j \leq 10, \quad j = 1, 2, \dots, 10
 \end{aligned}$$

The objective function  $f_4$  is quadratic and its globally optimal solution is as follows.

$$\begin{aligned}
 \mathbf{x}^* &= (2.171996, 2.363683, 8.773926, \\
 & 5.095984, 0.9906548, 1.430574, \\
 & 1.321644, 9.828726, 8.280092, 8.375927), \\
 f_4(\mathbf{x}^*) &= 24.3062091.
 \end{aligned}$$

For the optimal solution, six constraints are active.

**Problem #5**

$$\begin{aligned}
 \text{minimize } f_5(\mathbf{x}) &= (x_1 - 10)^3 + (x_2 - 20)^3 \\
 \text{subject to } & - (x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\
 & (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0
 \end{aligned}$$

$$\begin{aligned}
 & - 3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \\
 & 13 \leq x_1 \leq 100, \quad 0 \leq x_2 \leq 100
 \end{aligned}$$

The objective function  $f_5$  is polynomial and its globally optimal solution is as follows.

$$\mathbf{x}^* = (14.095, 0.84296), f_5(\mathbf{x}^*) = -6961.81381.$$

For the optimal solution, all constraints are active.

The properties of test problems are summarized in Table 1. The table shows the number of decision variables  $n$ , the type of the objective function, the number of linear constraints  $m_L$ , the number of nonlinear constraints  $m_N$ , the number of active constraints at the optimal solution  $m^*$ .

**Table 1. The Properties of Test Problems**

Problem	Type of $f$	$m_L$	$m_N$	$m^*$
#1	quadratic	9	0	6
#2	linear	3	3	6
#3	polynomial	0	4	2
#4	quadratic	3	5	6
#5	polynomial	0	3	3

We apply RGENOCOP III [8], GENOCOP V [9] and the proposed method (RGENOCOP V) to these problems and compare their results.

In this experiment, the parameters of genetic algorithms are set as: the number of individuals = 70, the maximal search generation number = 5000 and the number of trials = 10. Table 2 shows the results. In the table, RG III, GV and RG V mean RGENOCOP III, GENOCOP V and RGENOCOP V, respectively.

Next, we apply these methods to the following nonconvex programming problem.

**Problems #6**

$$\begin{aligned}
 \text{minimize } f_4(\mathbf{x}) &= x_1^3 + (x_2 - 5)^2 + 3(x_3 - 9)^2 \\
 & - 12x_3 + 2x_4^3 + 4x_5^2 + (x_6 - 5)^2 \\
 & + (x_6 - 5)^2 - 6x_7^2 + 3(x_7 - 2)x_8^2 \\
 & - x_9x_{10} + 4x_9^3 + 5x_1x_3 - 3x_1x_7 \\
 & \quad + 2x_8x_{10} \\
 \text{subject to } & 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 \\
 & \quad + 2x_5x_6x_8 - 120 \leq 0 \\
 & 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\
 & x_1^2 + 2(x_1 - 2)^2 - 2x_1x_2 + 14x_5 \\
 & \quad + 6x_5x_6 \leq 0 \\
 & 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2x_6x_8 \\
 & \quad - 30 \leq 0
 \end{aligned}$$

$$5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$$

$$- 3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$$

$$4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 105$$

$$10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$$

$$- 8x_1 + 2x_2 + 5x_7 - 2x_{10} \leq 12$$

$$0 \leq x_j \leq 10, j = 1, 2, \dots, 10$$

**Table 2. Results for Five Test Problems**

Prob.		Objective Function Value		
		RG III	G V	RG V
#1	Best	-14.89	-15.00	-15.00
	Average	-14.89	-9.68	-12.88
	Worst	-14.89	-2.15	-9.50
	Optimal	-15.00		
#2	Best	7129.00	7255.54	7068.44
	Average	7508.32	8559.57	7182.63
	Worst	7796.21	11597.24	7273.53
	Optimal	7049.33		
#3	Best	680.63	680.65	680.63
	Average	680.63	680.67	680.85
	Worst	680.63	680.69	680.98
	Optimal	680.63		
#4	Best	25.57	24.62	24.78
	Average	28.23	27.14	25.31
	Worst	35.25	35.79	25.61
	Optimal	24.30		
#5	Best	-6961.81	-6873.59	-6961.81
	Average	-6961.80	-6787.49	-6961.79
	Worst	-6961.80	-6614.13	-6961.77
	Optimal	-6961.81		

The globally optimal solution to this problem is unknown. In the application to this problem, the parameters of genetic algorithms are set as: the number of individuals = 70, the maximal search generation number = 5000 and the number of trials = 10. Table 3 shows the results.

**Table 3. Results for Problem #6**

Prob.		Objective Function Value		
		RG III	G V	RG V
#6	Best	9.12	9.31	8.87
	Average	11.50	21.38	10.14
	Worst	13.84	40.05	13.19

From Tables 2 and 3, RGENOCOP V proposed in this paper gives better results than those by GENOCOP V [9] with respect to accuracy and precision. In the comparison of results by RGENOCOP V with those by RGENOCOP III [8], RGENOCOP V can obtain better results for #2, #4 and #6, while RGENOCOP III is superior for #1, #3 and #5. Table 4 summarizes the results of Tables 2 and 3.

Table 4 indicates that the proposed RGENOCOP V was more accurate and more precise than other methods on average for problems in numerical experiments. Furthermore, we show the computation time in Table 5. From Table 5, with respect to the computation time, there is no serious difference between RGENOCOP V, GENOCOP V [9] and RGENOCOP III [8]. In addition, the transition of objective function values with the progress of generation in a trial is shown in Table 6 and Fig. (4).

**Table 4. The Summary of the Results of Tables 2 and 3**

Prob.	Criterion	Best of RG III, G V, RG V
#1	Accuracy	G V, RG V
	Precision	RG III
#2	Accuracy	RG V
	Precision	RG V
#3	Accuracy	RG III, RG V
	Precision	RG III
#4	Accuracy	G V
	Precision	RG V
#5	Accuracy	RG III, RG V
	Precision	RG III
#6	Accuracy	RG V
	Precision	RG V

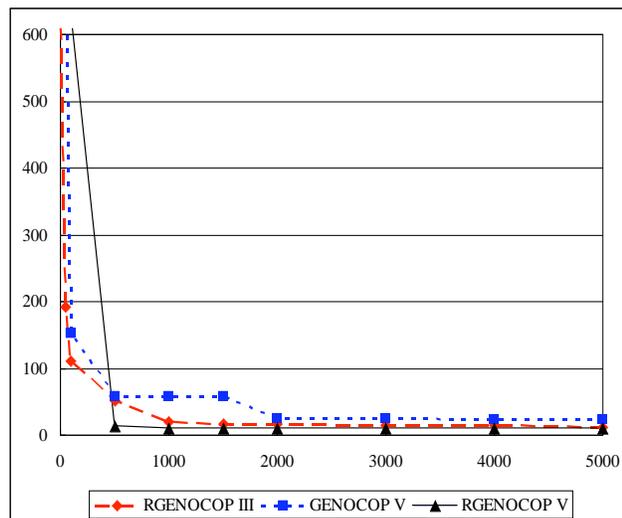
**Table 5. Computation Time**

Prob.	Computation Time (Sec.)		
	RG III	G V	RG V
#1	315	343	339
#2	209	310	291
#3	144	214	131
#4	174	432	368
#5	60	229	185
#6	219	354	386

In Table 6, it is shown that the improvement of solutions obtained by RGENOCOP III [8] and that by GENOCOP V [9] are slow, while solutions obtained by RGENOCOP V are rapidly improved. From this result, RGENOCOP V could find better solutions in earlier generations, i.e., in shorter time than RGENOCOP III and GENOCOP V.

**Table 6.** The Transition of Objective Function Values with the Progress of Generation in a Trial

Generation	Objective Function Value		
	RG III	G V	RG V
1	651.49	2641.93	929.38
50	191.12	860.18	726.49
100	111.26	152.53	638.70
500	51.75	57.25	12.84
1000	19.81	57.25	9.92
1500	17.04	57.25	9.75
2000	16.49	24.14	9.56
3000	15.23	24.00	9.52
4000	14.82	23.57	9.52
5000	12.14	23.57	9.52

**Fig. (4).** The transition of objective function values with the progress of generation in a trial.

## CONCLUSIONS

In this paper, focusing on nonconvex nonlinear programming problems, we considered a new solution method based on a floating-point genetic algorithm, GENOCOP V [9].

Unfortunately, in GENOCOP V, there exist two shortcomings: the difficulty of generating initial feasible solutions and the stop around local optimal solutions. For dealing with the first shortcoming, we incorporated an efficient method to generate initial feasible solutions by solving an optimization problem to minimize the degree of violation of constraints into it. In addition, for the second shortcoming, we proposed the procedure of updating basepoint solutions of the homomorphous mapping. Furthermore, in order to show the efficiency of the new method proposed in this paper, called RGENOCOP V, which adopts two improvements mentioned above, we applied it with existing methods (RGENOCOP III [8] and GENOCOP V [9]) to several numerical examples. Since the proposed RGENOCOP V was more accurate and more precise than other methods on average in numerical experiments, it seems a promising approximate solution method for nonconvex nonlinear programming problems. In near future, we will report the application of RGENOCOP V into multiobjective nonconvex nonlinear programming problems.

## REFERENCES

- [1] M. C. Biggs, Constrained minimization using recursive quadratic programming. in *Towards Global Optimization*, L.C.W. Dixon and G.P. Szego, Eds. Amsterdam: North-Holland, 1975.
- [2] J. Abadie and J. Carpenter. Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints in *Optimization*. R. Fletcher, Ed. New York: Academic Press, 1969.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.
- [4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [5] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York, NY: John Wiley & Sons, 1997.
- [6] M. Sakawa, *Genetic Algorithms and Fuzzy Multiobjective Optimization*. Dordrecht, Netherlands: Kluwer Academic Publishers, 2001.
- [7] Z. Michalewicz and G. Nazhiyath, "Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints." *Proc. 2nd IEEE Int. Conf. on Evol. Comput.*, pp. 647-651, 1995.
- [8] M. Sakawa and K. Yauchi, "An interactive fuzzy satisficing method for multiobjective nonconvex programming problems with fuzzy numbers through coevolutionary genetic algorithms." *IEEE Trans. Syst., Man, and Cybern. B*, vol. 31, no. 3, pp. 459-467, June 2001.
- [9] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization." *Evol. Comput.*, vol. 7, no. 1, pp. 19-44, April 1999.
- [10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York, NY: Springer-Verlag, 1992.