



Title	A study of operator and parameter choices in non-revisiting genetic algorithm
Author(s)	Yuen, SY; Chow, CK
Citation	The 2009 IEEE Congress on Evolutionary Computation (CEC 2009), Trondheim, Norway, 18-21 May 2009. In IEEE Transactions on Evolutionary Computation, 2009, p. 2977-2984
Issued Date	2009
URL	http://hdl.handle.net/10722/196711
Rights	Creative Commons: Attribution 3.0 Hong Kong License

A Study of Operator and Parameter Choices in Non-Revisiting Genetic Algorithm

Shiu Yin Yuen and Chi Kin Chow

Abstract— We study empirically the effects of operator and parameter choices on the performance of the non-revisiting genetic algorithm (NrGA). For a suite of 14 benchmark functions that include both uni-modal and multi-modal functions, it is found that NrGA is insensitive to the axis resolution of the problem, which is a good feature. From the empirical experiments, for operators, it is found that crossover is an essential operator for NrGA, and the best crossover operator is uniform crossover, while the best selection operator is elitist selection. For parameters, a small population, with a population size strictly larger than 1, should be used; the performance is monotonically increasing with crossover rate and the optimal crossover rate is 0.5. The results of this paper provide empirical guidelines for operator designs and parameter settings of NrGA.

I. INTRODUCTION

Guidelines on operator choices and parameter values are valuable information for industry practitioners and researchers who wish to use an algorithm to solve their research problems. In this paper, we empirically study the effects of applying different operators and parameter values to the non-revisiting genetic algorithm (NrGA)[1, 2].

The original motivation of NrGA aims to answer a fundamental question: can a stochastic search algorithm – in this case a GA – be designed such that the search will never revisit a location that has been visited before? For deterministic search problems, the answer is simple because one can always search in a particular order. For stochastic search, in general, one needs to remember every visited search positions. Is that worthwhile or feasible?

The answer is definitely yes for applications involving expensive or time consuming fitness evaluations, or both. Many day to day engineering applications fall into this category. For such applications, a fitness evaluation may take from 1 second to 1 day. Some examples are antenna design [3], heating, ventilating and air conditioning (HVAC) engineering [4] and three dimensional object registration in computer vision [5]. In such application, it is only natural to store all fitness evaluation results. Since the total number of evaluations is limited, the memory required is acceptable.

Shiu Yin Yuen is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China (E-mail: kelviny.ee@cityu.edu.hk).

Chi Kin Chow is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China (E-mail: chowchi@cityu.edu.hk).

The work described in this paper was supported by CityU Research Grant (7002304).

This is especially so for future evolutionary algorithms as memory available is increasing rapidly due to Moore's law.

In addition to storing all visited solutions, NrGA aims at storing and organizing the *entire search history* in a way to make *intelligent* search decisions actively. A first demonstration, as used in NrGA, is to use a binary space partitioning (BSP) tree in computational geometry [6] to organize the visited solutions. This gives naturally an adaptive mutation operator *that has no parameter*. Consequently, the NrGA has very few parameters. The following are its parameters: (1) axis resolution parameter d ; this is the number of discrete levels for each gene; (2) μ and λ for the $(\mu+\lambda)$ selection; and (3) crossover rate r_x . d is application dependent. In [2], $\mu=100$, $\lambda=200$, $r_x=0.5$.

In [2], the operator choices are: (1) elitist $(\mu+\lambda)$ selection and (2) uniform crossover.

An optional background mutation operator of $1/D$ type may also be added, where D is the problem dimension. It is turned on in [1] and is turned off in [2] (to better illustrate the performance gain due to the parameter-less adaptive mutation.)

The NrGA design is illustrated in Figure 1. As seen, it can be appreciated that it is just a simple GA coupled with an adaptive mutation module.

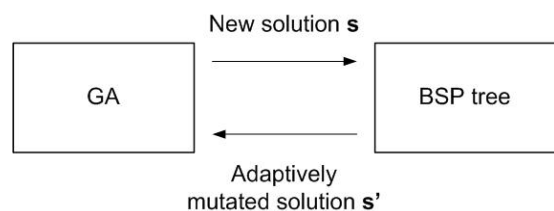


Figure 1. Communication between GA and the BSP tree

NrGA was compared with (1) real coded GA (RC-GA); (2) Covariance Matrix Adaptation Evolution Strategy (CMA-ES) – a state of the art adaptive mutation method; (3) canonical GA (CGA); (4) GA with a simple diversity mechanism (Div-GA) and (5) three forms of enhanced particle swarm optimization (DPSO, SEPSO and PSOMS). The test was performed on 19 well known benchmark functions in a total 64 test cases. It is observed that NrGA outperforms the other test algorithms: NrGA ranks first (or joint first) in 39 and is second in 13 out of the 64 cases. For full details of NrGA, please refer to [2].

Particularly worthy of mention is that NrGA outperforms CMA-ES, which is the choice method recommended by experts for adaptive mutation [7]. Note that in CMA-ES, the

parameter settings have been fine tuned by hand.

This motivates us to study why NrGA works. In particular, we wish to answer the question: what is the influence of operator and parameter choices and settings in NrGA designs? Is there a recommended design for NrGA? In this paper, we study the effects of operator choices and parameter settings in NrGA: We did control experiments on 1) population size $\mu = \lambda$. (λ is set equal to μ .) 2) axis resolution d , 3) crossover rate r_x , 4) crossover operator and 5) selection scheme. We shall draw some conclusions based on the results of these empirical experiments.

The rest of this paper is organized as follows: Section II explains the simulation setup. Section III reports the results on varying population size. Section IV reports the results of changing the axis resolution. Section V reports the effect of varying the crossover rate. Section VI reports the effects of different crossover operators. Section VII reports on using different selection schemes. Section VIII gives the conclusion.

II. SIMULATION SETUP

In this paper, we qualify the performance of NrGA by its best fitness. The performance of NrGA for a particular setting at a particular objective function is concluded by the averaged best fitness of 100 independent runs.

A. Test functions

A real valued function set $\mathbf{F} = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{14}(\mathbf{x})\}$ consisting of 14 well known benchmark functions are employed. The test functions, which can be found in [8], are as follows:

1. Sphere function
2. Schwefel's problem 2.22
3. Schwefel's problem 1.2
4. Schwefel's problem 2.21
5. Generalized Rosenbrock function
6. Quartic function
7. Generalized Rastrigin function
8. Generalized Griewank function
9. Generalized Schwefel's problem 2.26
10. Ackley function
11. Shekel's Foxholes
12. Six-Hump Camel-Back function
13. Branin function
14. Goldstein-Price function

The mathematical forms, the search space and the optima of these functions are given Table 5.

The first six functions are uni-modal functions; the remaining eight are multi-modal functions designed with a considerable amount of local minima. The dimensions of the first ten functions are adjustable while the dimensions of $f_{11} - f_{14}$ are fixed at two. All functions with the exceptions of $f_9, f_{11},$

f_{12} and f_{13} , have the global minimum at the origin or very close to the origin. Simulations are carried out to find the global *minimum* of each function.

B. Basic setting of NrGA

In this paper, we examine the sensitivities of six operators /parameters of NrGA including: 1) population size μ , 2) axis resolution d , 3) crossover rate r_x , 4) crossover operator $C(\cdot)$, and 5) selection scheme $S(\cdot)$. In each of the sensitivity analyses, only one of the operators /parameters is varied and we observe the influence to the averaged best fitness value. In all the experiments, NrGA has the following settings throughout this paper unless specified otherwise:

- Population size $\mu = 100$
- Numbers of generations N_g : 400 for the first ten test functions and 40 for the last four test functions.
- Axis resolution d : 100 for the first ten functions and 2000 for the last four test functions.
- Number of dimensions D : 30 for the first ten test functions and 2 for the last four test functions.
- Crossover operator $C(\cdot) =$ uniform crossover
- Crossover rate $r_x = 0.5$
- Selection scheme $S(\cdot) =$ elitism selection

For example, in section IV, the parameters except population size μ is assigned as: $d = 100$, $N_g = 400$ or 40, $C(\cdot)$ is uniform crossover, $r_x = 0.5$, $S(\cdot) =$ elitism selection and $D = 30$. Similarly, in section V, the parameters except axis resolution d are assigned as: $\mu = 100$, $N_g = 400$ or 40, $C(\cdot)$ is uniform crossover, $r_x = 0.5$, $S(\cdot) =$ elitism selection and $D = 30$.

III. POPULATION SIZE

In this experiment, we examine the performance of NrGA for different population sizes. For test functions $f_1 - f_{10}$, the population size is varied from 2 to 20,000. Table 1 lists the combinations of population size μ against number of generation N_g . The total number of evaluations for $f_1 - f_{10}$ is kept constant at 40,000.

For test functions $f_{11} - f_{14}$, the population size is varied from 2 to 2,000. Table 2 lists the combinations of μ and N_g . The total number of evaluations for $f_{11} - f_{14}$ is kept constant at 2,000.

Figure 2 shows the averaged best fitness value against population size for $f_1 - f_{14}$. The general trend is that the averaged best fitness value increases with population size. So the population size should be kept small. For $f_1 - f_{10}$, the optimal population size is 2; for $f_{11} - f_{14}$, the optimal population size is small. Note that the population size cannot be 1, otherwise the crossover operator degenerates and there is a drastic drop in performance (see section V).

TABLE 1. THE COMBINATIONS OF POPULATION SIZE AND NUMBER OF GENERATIONS FOR $f_1 - f_{10}$ IN EXPERIMENT 1.

μ	N_g	μ	N_g	μ	N_g

2	20,000	80	500	1,250	32
4	10,000	100	400	1,600	25
5	8,000	125	320	2,000	20
8	5,000	160	250	2,500	16
10	4,000	200	200	4,000	10
16	2,500	250	160	5,000	8
20	2,000	320	125	8,000	5
25	1,600	400	100	10,000	4
32	1,250	500	80	20,000	2
40	1,000	625	64		
50	800	800	50		
64	625	1,000	40		

TABLE 2. THE COMBINATIONS OF POPULATION SIZE AND NUMBER OF GENERATIONS FOR $f_{11} - f_{14}$ IN EXPERIMENT 1

μ	N_g	μ	N_g	μ	N_g
2	2,000	32	125	250	16
4	1,000	40	100	400	10
5	800	50	80	500	8
8	500	80	50	800	5
10	400	100	40	1,000	4
16	250	125	32	2,000	2
20	200	160	25		
25	160	200	20		

IV. AXIS RESOLUTION

Axis resolution d directly affects the precision of the optimum found by NrGA; the precision is raised by increasing the resolution. According to the adaptive mutation scheme of NrGA, the mutation step size depends on the size of the unvisited subspace rather than the resolution. Nonetheless, if the resolution is low, the chance of revisit and hence node pruning occurs more frequently. These will benefit NrGA in two ways: firstly, node pruning will simplify the BSP tree and the time for node search will be shorter. Also, NrGA can escape from a local optimum when the unvisited subspace of the local optimum is fully evaluated.

However, as the resolution increases, node pruning rarely occurs. As the mutation step size is independent of the resolution, the performance of NrGA should be invariant to the resolution when it is sufficiently high. This predicted performance is shown in Figure 3. When $d \leq 2^{13} = 8192$ (i.e. the usual range settings for GA), the performance is unchanged when d is varied. Interestingly, as the resolution increases from 2^{15} to 2^{20} , i.e. the resolution is close to being continuous, the performance is also unchanged.

V. CROSSOVER RATE

In this experiment, we examine the performance of NrGA for crossover rate r_x at 0.0, 0.05, ..., 0.5. TABLE 6 and TABLE 7 show the averaged best fitness values against crossover rate for $f_1 - f_{14}$. Seen from the figures, the averaged best fitness monotonically decreases with increases in crossover rate. The results suggest that the optimal crossover rate (using uniform crossover) is 0.5.

Comparing the first two rows of the tables, it can be observed that there is a drastic performance improvement when r_x increases from 0 to 0.05. This suggests that the crossover operator is an essential operator in NrGA.

VI. CROSSOVER OPERATOR

In this experiment, we examine the performance of NrGA for four crossover operators: 1) Uniform crossover, 2) 1-point crossover, 3) 2-point crossover and 4) Arithmetic crossover. For uniform crossover, the crossover rate is 0.5. For the 1-point and 2-point crossover operators, the crossover boundaries are randomly selected. For arithmetic crossover, the crossover weight is randomly selected between [0, 1]. Since the 2-point crossover operator involves two crossover boundaries, an individual must have at least three genes. So the dimension of the corresponding objective function should be at least three. Thus only the first ten test functions $f_1 - f_{10}$ are employed in this experiment.

Table 3 lists the averaged best fitness for the four crossover operators. The values inside the brackets indicate the standard deviations of the fitness values. The crossover operator with the smallest fitness value for each test function is in bold. Note that there is a considerable performance difference amongst the operators. This suggests that the performance of NrGA is sensitive to the crossover operator choice.

Seen from the table, the uniform crossover ranks first (or joint first) in 8 out of 10 test functions. Thus, it is suggested to employ the uniform crossover operator in NrGA when there is no knowledge about the selection of crossover operator. To illustrate the significance of the recommendation of uniform crossover, the confidence interval (in term of %) for t-test comparing the averaged best fitness values of the uniform crossover with the other three operators are listed. The confidence intervals of the uniform crossover compared to another operator is indicated by “-“ if the averaged best fitness values of both operators are equal.

The significance of the rank of the uniform crossover is summarized as follows: We are 99.95% confident that the uniform crossover is the best among the four operators in 4 test functions. For the remaining four functions that the uniform crossover ranks or jointly ranks first, one out of them is with 99.75% confidence interval, and the remaining three are with 95% confidence interval. From the above, we conclude that the recommendation of crossover operator is statistically significant.

TABLE 3. AVERAGED BEST FITNESS OF THE TEST FUNCTIONS FOR DIFFERENT CROSSOVER-OPERATORS

	Uniform	1-point	2-point	Arithmetic
f_1	0.000 (0.000) -	0.0108 (0.0601) 95%	0.000 (0.000) -	8.3768 (6.667) 99.95%
f_2	0.000 (0.000) -	0.516 (1.2179) 99.95%	0.124 (0.665) 95%	15.054 (3.176) 99.95%
f_3	2554.04 (1353.7626) -	6222.96 (2526.581) 99.95%	5325.88 (1956.1209) 99.95%	8156.6 (3155.758) 99.95%
f_4	35.28 (5.3853) -	64.16 (5.3865) 99.95%	61.58 (6.4516) 99.95%	19.1 (2.7622) 99.95%
f_5	463.1292 (653.95) -	4696.4448 (11442.63) 99.95%	1351.2924 (2921.70) 99.75%	194773.68 (119623.79) 99.95%
f_6	8.6445	16.4004	13.394	8.4447

	(0.5138)	(2.5817)	(2.0214)	(0.4907)
	-	99.95%	99.95%	99.50%
f_7	16.4178	69.9228	56.849	64.6156
	(5.7438)	(16.2481)	(14.8125)	(9.0986)
	-	99.95%	99.95%	99.95%
f_8	0.000	0.0569	0.003	16.6708
	(0.000)	(0.1365)	(0.0173)	(5.8675)
	-	99.95%	95%	99.95%
f_9	-13047.6325	-10752.4713	-11175.6634	-6294.315
	(227.4735)	(616.7379)	(548.9175)	(751.7627)
	-	99.95%	99.95%	99.95%
f_{10}	0.000	11.3348	5.1462	9.1697
	(0.000)	(4.5851)	(4.7577)	(1.0871)
	-	99.95%	99.95%	99.95%

VII. SELECTION SCHEME

In this experiment, we examine the performance of NrGA for three selection schemes:

1. Elitist selection (elitism) – Suppose $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,2,\dots,n}$ is the parent pool and $\mathbf{O} = \{\mathbf{o}_i\}_{i=1,2,\dots,n}$ is the offspring pool. Elitist selection chooses the n individual with the smallest fitness from the mixture of the pools \mathbf{X} and \mathbf{O} to form the new population.
2. Rank-based proportional selection (proportional) – Suppose $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,2,\dots,n}$ is the parent pool and $\mathbf{O} = \{\mathbf{o}_i\}_{i=1,2,\dots,n}$ is the offspring pool, $\mathbf{M} = \{\mathbf{m}_i\}_{i=1,2,\dots,2n} = \{\mathbf{X}, \mathbf{O}\}$ is the mixture of the pools \mathbf{X} and \mathbf{O} , the probability that $\mathbf{m}_j \in \mathbf{M}$ is chosen as the member of the new population is:

$$(2n - r_j + 1) \times \left(\sum_{i=1}^{2n} i \right)^{-1}$$

where r_j is the rank of \mathbf{m}_j in \mathbf{M} .

3. Tournament selection (tournament) - Suppose $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,2,\dots,n}$ is the parent pool, every individual \mathbf{o}_k in the offspring pool $\mathbf{O} = \{\mathbf{o}_i\}_{i=1,2,\dots,n}$ is generated as follows: Given a tournament size t , we first randomly choose t individuals from \mathbf{X} and form the sub-pool $S = \{\mathbf{s}_j\}_{j=1,2,\dots,t}$ where $\mathbf{s}_j \in \mathbf{X}$ for all j . The probability that \mathbf{o}_k is assigned as \mathbf{s}_j is:

$$g_k \times \left(\sum_{j=1}^t g_j \right)^{-1}$$

where $g_k = f(\mathbf{s}_j)^{-1}$ and $f(\cdot)$ is the objective function. In this experiment, the tournament size is chosen to be 10.

Table 4 lists the averaged best fitness for the three selection schemes. The values inside the brackets indicate the standard deviations of the fitness values. The selection scheme with the smallest fitness value for each test function is in bold. Seen from the table, the elitist selection ranks first for eleven test functions; the rank-based proportional selection performs best for the other three test functions.

Thus it is suggested to employ the elitism selection in NrGA. To illustrate the significance of the recommendation of this selection scheme, the confidence interval (in term of %) for t-test comparing the averaged best fitness values of the

elitism selection with the other selection schemes are listed.

The significance of the rank of the elitism selection is summarized as follows: We are 99.95% confident that the uniform crossover is the best among the four operators in seven test functions. For the remaining four functions that the elitism selection ranks first, one out of them is with 97.5% confidence interval, and the remaining three test functions are with 80% confidence interval. From the above, we conclude that the recommendation of selection scheme is statistically significant.

TABLE 4. AVERAGED BEST FITNESS OF THE TEST FUNCTIONS FOR DIFFERENT SELECTION SCHEMES

	Elitism	Proportional	Tournament
f_1	0.000	0.0092	0.028
	(0.000)	(0.0168)	(0.0421)
	-	99.95%	99.95%
f_2	0.000	0.024	86.66
	(0.000)	(0.065)	(13.9124)
	-	99.95%	99.95%
f_3	2431.76	6184.16	33120.76
	(1090.6183)	(2268.1166)	(8424.3433)
	-	99.95%	99.95%
f_4	35.84	30.64	44.1
	(5.9778)	(5.2603)	(7.8198)
	-	99.95%	99.95%
f_5	513.99	725.7024	2829.114
	(713.6914)	(777.2359)	(4342.3787)
	-	97.5%	99.95%
f_6	8.7656	8.5717	10.5206
	(0.4943)	(0.4579)	(0.969)
	-	99.5%	99.95%
f_7	17.5224	28.9291	56.4179
	(4.9273)	(7.8545)	(12.0177)
	-	99.95%	99.95%
f_8	0.000	0.0518	0.1247
	(0.000)	(0.0634)	(0.0932)
	-	99.95%	99.95%
f_9	-13034.4643	-12623.4954	-11432.4903
	(219.4799)	(337.9245)	(503.3919)
	-	99.95%	99.95%
f_{10}	0.000	0.0849	5.6249
	(0.000)	(0.2105)	(4.6608)
	-	99.95%	99.95%
f_{11}	2.3324	2.7103	3.0519
	(2.0394)	(2.9352)	(2.4955)
	-	80%	97.5%
f_{12}	-1.0277	-1.0221	-0.859
	(0.0245)	(0.057)	(0.2463)
	-	80%	99.95%
f_{13}	0.406	0.4143	0.7462
	(0.0568)	(0.0653)	(0.6431)
	-	80%	99.95%
f_{14}	3.871	3.7481	10.2517
	(3.9503)	(3.6385)	(8.5555)
	-	< 50%	99.95%

VIII. CONCLUSION

Non-revisiting genetic algorithm (NrGA) is a promising new evolutionary algorithm. It has superior performance compared with some well known evolutionary algorithms. Its merits are (1) simple design, with few parameters; (2) the use of the *entire* search history; (3) the ability to make intelligent decisions from *all* past knowledge gained. It features a

powerful parameter-less adaptive mutation operator, which is parameter free but performs impressively well. This motivates us to investigate why the idea works.

In this paper, we report empirical investigations on the influence of operator and parameter choices on NrGA. It is found that (1) a small population, roughly from 2 to 200, gives the best performance; (2) the performance is stable and constant when the axis resolution parameter varies (a) within the ranges normally used in GA applications and (b) when the resolution parameter is close to continuous [9]; (3) crossover is an essential operator and should be used; (4) the best crossover operator is the uniform crossover; (5) the best crossover rate is 0.5, the maximum rate possible; and (6) the best selection operator is the elitist selection operator.

The results of this paper contributes to the evolutionary computation community in three ways: (1) it shows that NrGA is insensitive to axis resolution, which makes it attractive to apply to diverse practical application; (2) it gives a set of empirical guidelines on operator and parameter choices of NrGA, which are useful information for industry practitioners interested in applying the NrGA to their engineering applications; and lastly, (3) it gives some data to theorists who wish to investigate a rigorous theoretical model of NrGA.

Instead of fixed operators and parameters, new schemes for operator and parameter control [7] for NrGA can be investigated in future, notwithstanding that operator and parameter control is a very hard problem in general. It is interesting to note in passing that the parameter-less adaptive mutation in NrGA constitutes a form of parameter control that

adaptively discovers its control mutation operators and mutation parameters from the search process itself.

REFERENCES

- [1] S. Y. Yuen and C. K. Chow, "A Non-revisiting Genetic Algorithm", in *Proc. IEEE CEC*, pp. 4583 – 4590, 2007.
- [2] S.Y. Yuen and C.K. Chow, "A Genetic algorithm that adaptively mutates and never revisits," *IEEE Transactions on Evolutionary Computation*, to be published.
- [3] Q. X. Chu, K. F. Chan and C. H. Chan, "Parallel FDTD analysis of active integrated antenna array," *Microwave and Optical Technology Letters*, vol. 34, pp. 317-319, 2002.
- [4] K. F. Fong, V. I. Hanby and T. T. Chow, "HVAC system optimization for energy management by evolutionary programming," *Energy and Buildings*, vol. 38, pp. 220-231, 2006.
- [5] C. K. Chow, H. T. Tsui and T. Lee, "Surface registration using a dynamic genetic algorithm," *Pattern Recognition*, vol. 37, no. 1, pp. 105-117, 2004.
- [6] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational geometry, algorithms and applications*, 2nd Ed., 2000.
- [7] F.G. Lobo, C.F. Lima and Z. Michalewicz (Eds.), *Parameter setting in evolutionary algorithms*, Springer, 2007.
- [8] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," *IEEE Trans. Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [9] S.Y. Yuen and C.K. Chow, "Continuous non-revisiting genetic algorithm," in *Proc. IEEE CEC*, 2009.

TABLE 5 DETAILS OF THE FOURTEEN TEST FUCNTIONS.

<i>test function</i>	<i>mathematical form</i>	<i>range</i>	<i>optimum</i>
1. Sphere function	$f_1(\mathbf{x}) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	$[0, 0, \dots, 0]$
2. Schwefel's problem 2.22	$f_2(\mathbf{x}) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-100, 100]^D$	$[0, 0, \dots, 0]$
3. Schwefel's problem 1.2	$f_3(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$	$[0, 0, \dots, 0]$
4. Schwefel's problem 2.21	$f_4(\mathbf{x}) = \max_{i \in \{1, D\}} x_i $	$[-100, 100]^D$	$[0, 0, \dots, 0]$
5. Generalized Rosenbrock function	$f_5(\mathbf{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-29, 31]^D$	$[1, 1, \dots, 1]$
6. Quartic function	$f_6(\mathbf{x}) = \sum_{i=1}^D ix^4 + \text{random}[0,1]$	$[-1.28, 1.25]^D$	$[0, 0, \dots, 0]$
Note: This is a noisy fitness function. There is a random measurement noise in each fitness evaluation.			
7. Generalized Rastrigin function	$f_7(\mathbf{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	$[0, 0, \dots, 0]$
8. Generalized Griewank function	$f_8(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \frac{x_i}{\sqrt{i}} + 1$	$[-600, 600]^D$	$[0, 0, \dots, 0]$
9. Generalized Schwefel's problem 2.26	$f_9(\mathbf{x}) = -\sum_{i=1}^D x_i \sin \sqrt{ x_i }$	$[-500, 500]^D$	$[420.9687, \dots, 420.9687]$
10. Ackley function	$f_{10}(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e$	$[-32, 32]^D$	$[0, 0, \dots, 0]$

11. Shekel's Foxholes function

$$f_{11}(\mathbf{x}) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{i,j})^6} \right]^{-1} \quad \text{where}$$

$$\{a_{i,j}\} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$$

[-98, 34]² [-32, 32]

12. Six-Hump Camel-Back function

$$f_{12}(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

[-4.91017, 5.0893] × [0.08983, -0.7126] and [-0.08983, 0.7126]

13. Branin function

$$f_{13}(\mathbf{x}) = (x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$$

[-8.142, 6.858] × [-12.275, 2.725] [-3.142, 12.275], [3.142, 2.275], [9.425, 2.425]

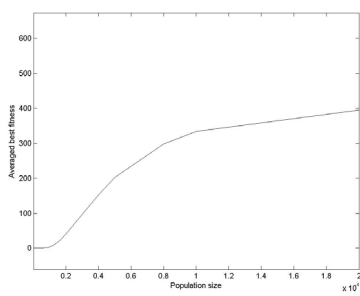
14. Goldstein-Price function

$$f_{14}(\mathbf{x}) = g(\mathbf{x}) \times h(\mathbf{x})$$

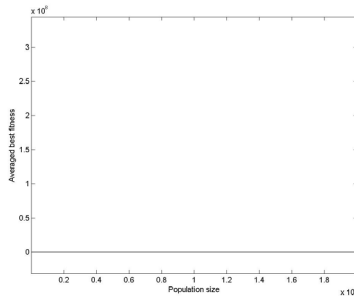
$$g(\mathbf{x}) = 1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 + 6x_1x_2 + 3x_2^2)$$

$$h(\mathbf{x}) = 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$$

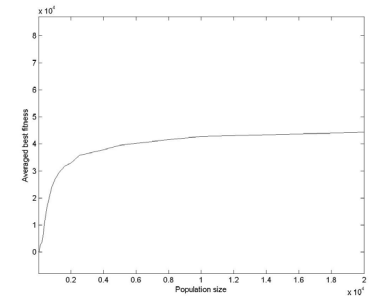
[-2, 2] × [-3, 1] [0, -1]



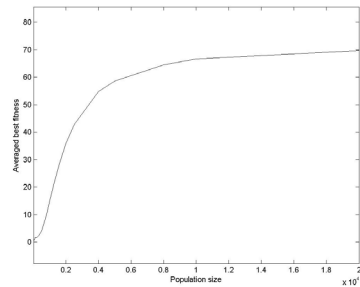
f_1



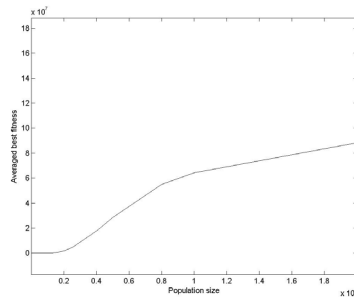
f_2



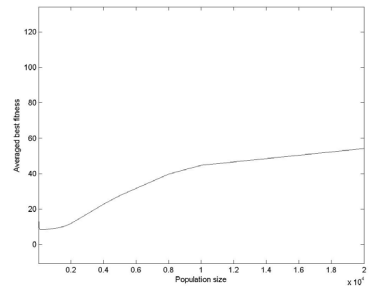
f_3



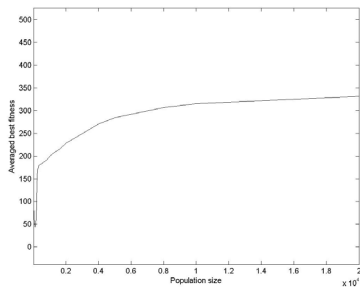
f_4



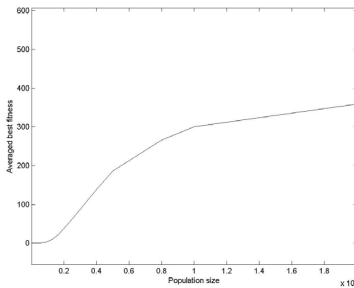
f_5



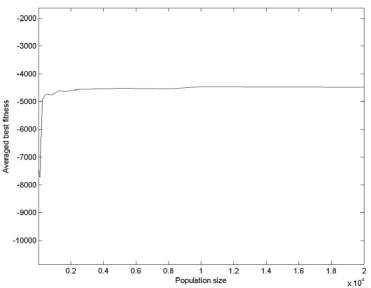
f_6



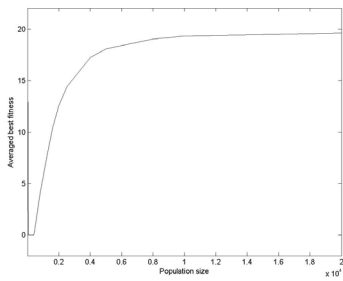
f_7



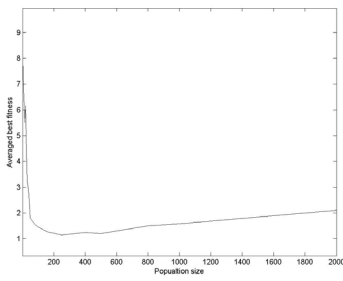
f_8



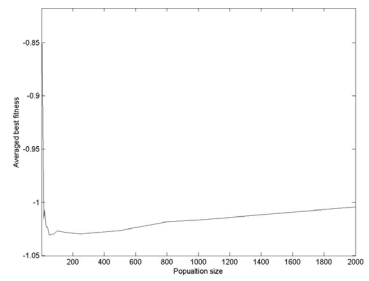
f_9



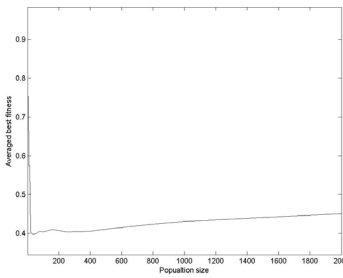
f_{10}



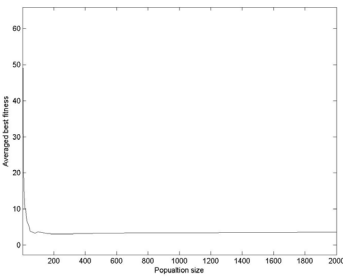
f_{11}



f_{12}

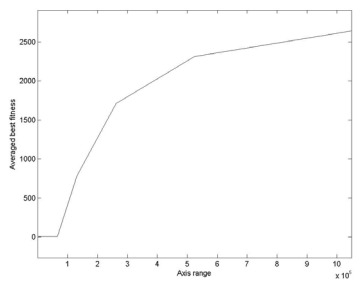


f_{13}

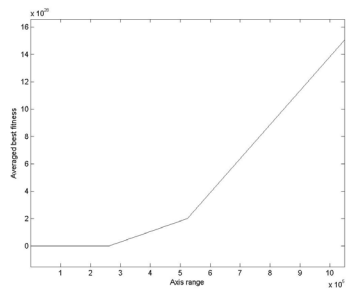


f_{14}

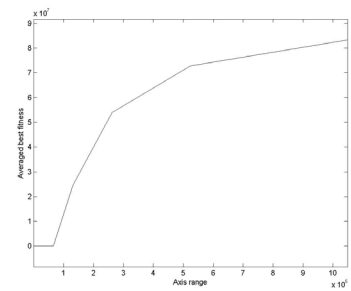
Figure 2. Averaged best fitness against population size for test functions $f_1 - f_{14}$



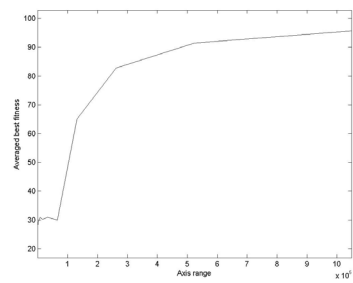
f_1



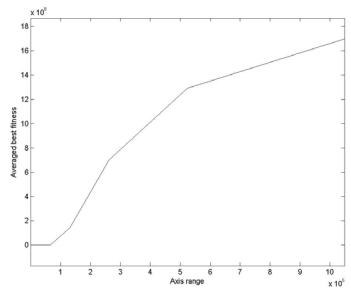
f_2



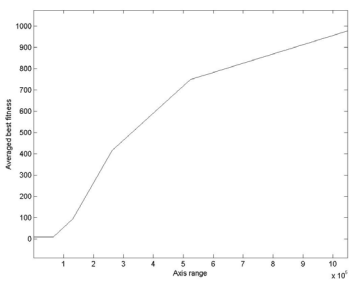
f_3



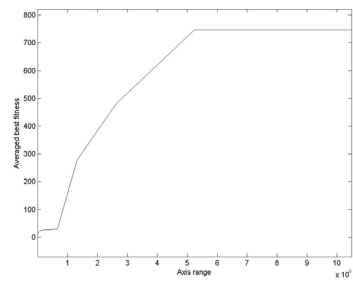
f_4



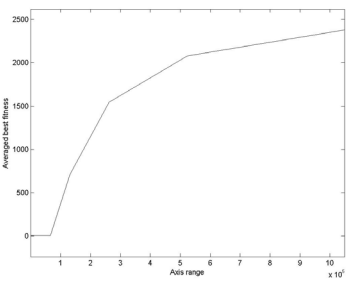
f_5



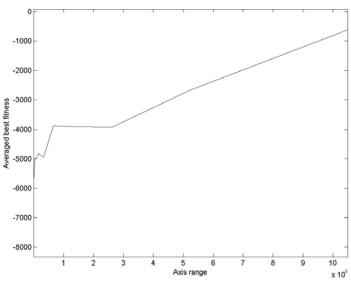
f_6



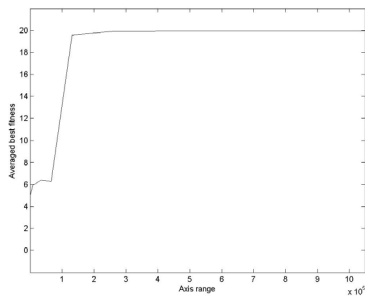
f_7



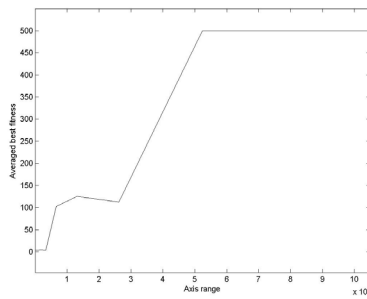
f_8



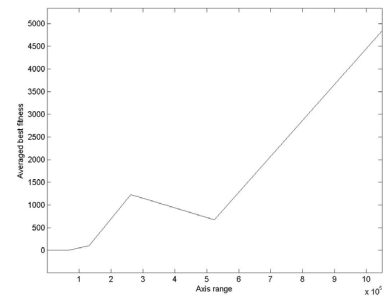
f_9



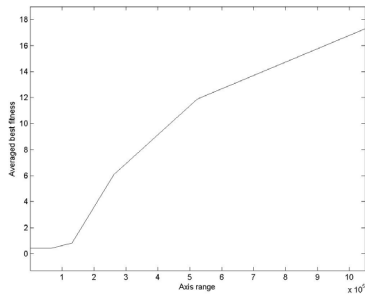
f_{10}



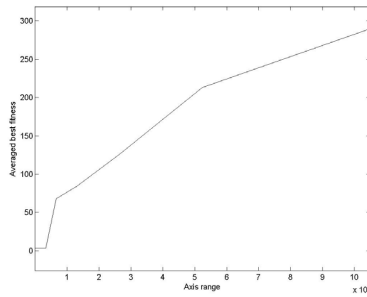
f_{11}



f_{12}



f_{13}



f_{14}

Figure 3. Averaged best fitness against axis resolution for test functions $f_1 - f_{14}$

TABLE 6. AVERAGED BEST FITNESS AGAINST CROSSOVER RATE FOR TEST FUNCTIONS $f_1 - f_7$

r_x	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0.00	177.81	61.08	24575.84	75.14	14732747.64	36.16	169.52
0.05	0.00	0.75	8523.12	64.60	5937.73	15.36	61.39
0.10	0.00	0.00	5986.56	58.56	816.94	11.55	39.55
0.15	0.00	0.00	4937.00	52.10	467.86	9.95	29.31
0.20	0.00	0.00	4121.60	48.62	711.10	9.38	24.14
0.25	0.00	0.00	3699.92	43.64	578.92	9.05	21.23
0.30	0.00	0.00	3516.40	40.04	679.11	8.78	20.32
0.35	0.00	0.00	3034.04	39.28	753.35	8.70	17.93
0.40	0.00	0.00	2777.20	36.62	648.40	8.75	17.61
0.45	0.00	0.00	2671.68	35.76	634.44	8.67	16.61
0.50	0.00	0.00	2528.44	35.96	583.29	8.67	16.41

TABLE 7. AVERAGED BEST FITNESS AGAINST CROSSOVER RATE FOR TEST FUNCTIONS $f_8 - f_{14}$

r_x	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
0.00	158.77	-8097.14	19.61	7.08	-0.98	0.47	7.79
0.05	0.09	-11013.04	7.35	4.30	-1.01	0.41	5.40
0.10	0.00	-12016.78	0.50	2.72	-1.03	0.40	4.25
0.15	0.00	-12518.36	0.00	2.88	-1.03	0.40	3.67
0.20	0.00	-12763.54	0.00	2.26	-1.03	0.40	3.16
0.25	0.00	-12856.69	0.00	2.21	-1.03	0.40	3.89
0.30	0.00	-12981.12	0.00	2.14	-1.03	0.40	3.43
0.35	0.00	-13011.61	0.00	2.53	-1.03	0.40	3.18
0.40	0.00	-13011.31	0.00	2.11	-1.03	0.40	3.83
0.45	0.00	-13088.86	0.00	2.22	-1.03	0.40	3.52
0.50	0.00	-12965.46	0.00	2.30	-1.03	0.40	3.45