

# An Adaptive Restarting Genetic Algorithm for Global Optimization

Son Duy Dao, Kazem Abhary, and Romeo Marian

**Abstract**—Genetic Algorithm (GA) is a popular stochastic optimization technique, often used to solve complex large scale optimization problems in many fields. Enhancing the search capability of GA is always desirable. In this paper, an innovative GA, called adaptive restarting GA, is developed to improve the global search capability of the algorithm. With an adaptive restarting procedure and elite chromosome strategy, the proposed GA is capable of jumping out of the local optima and finding the global optimum with very high success probability. Two benchmark functions are used to demonstrate the outperformance of the proposed GA, in comparison with five other algorithms available in global optimization literature, including the traditional GA.

**Index Terms**—Genetic algorithm, adaptive restarting procedure, elite chromosome, global optimization

## I. INTRODUCTION

GLOBAL optimization, referred as finding the best solution to an optimization problem which may have multiple local optima, is desirable in many fields such as engineering design, computer science, operations research, biomedicine, computational chemistry, etc. [1-3]. Generally, there are two classes of global optimization methods, namely, deterministic methods and stochastic methods [4]. Each class has strengths and weaknesses. Deterministic methods can guarantee the global optimal solutions for certain problems; however, they may fail when coping with black-box and large scale problems. Stochastic methods, on the other hand, are capable of working with any kind of problem but they have weak theoretical guarantee of global solutions. Stochastic methods only provide the global optimal solutions with guarantee in probability which will approach 1 in infinite time [5, 6]. Nevertheless, there is no algorithm capable of solving general global optimization problem with certainty in finite time [7].

Genetic Algorithm (GA) is a popular stochastic optimization method, often used to solve complex large scale problems [8]. GA has several advantages such as capability of handling large search space, ability to work with both continuous and discrete variables, flexibility in defining quality measures and constraints, capability of

providing multiple optimal/good solutions, and great potential for applying parallel computing techniques to shorten the processing time [9]. However, like the other stochastic optimization techniques, GA provides a weak guarantee of the global optimal solution; this guarantee is in probabilistic form only and it will decrease when the problem size increases [10]. Having a GA capable of finding the global optimal solution with high success probability is always desirable. In this paper, an innovative algorithm structure allowing GA to restart the searching process if it is stuck in local optima is proposed to improve the success probability of achieving the global optimal solution.

## II. LITERATURE REVIEW

GA is a bio-inspired optimization technique, first introduced by Holland [11]. GA has five main components, namely, chromosome encoding, crossover, mutation, evaluation and selection. Traditional structure of GA is shown in Fig. 1 and it is generally described as follows: “Genetic algorithm ... starts with an initial set of random solutions called population. Each individual in the population is called a chromosome representing a solution to the problem at hand... The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated using some measures of fitness. To create the next generation, new chromosome, called offspring, are formed by either (a) merging two chromosomes from current generation using a crossover operator or (b) modifying a chromosome using a mutation operator. A new generation is formed by (a) selecting, according to the fitness values, some of the parents and offspring and (b) rejecting others so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum or suboptimal solution to the problem” [12].

Since being introduced in 1975, a lot of developments in the five components of GA have been made, for example, real encoding [13], variable length chromosome [14], multi-point crossover [15], adaptive directed mutation [16], uniform/non-uniform mutation [17], multi-objective evaluation [18], tournament selection [19], dynamic ranking selection [20], etc. In addition, the traditional structure of GA has been improved to so called hybrid GA in some research, where other heuristic algorithms such as Hill Climbing, Simulated Annealing, Ant Colony Optimization, Particle Swarm Optimization, etc. were integrated into the GA [21-

Manuscript received: July 01, 2015; revised: July 21, 2015. This work was supported by Australian Government in the form of Endeavour Postgraduate Scholarship.

Mr Son Duy Dao is a PhD student at School of Engineering, University of South Australia, Australia (corresponding author to provide e-mail: son.dao@mymail.unisa.edu.au).

Prof. Kazem Abhary is with School of Engineering, University of South Australia, Australia (e-mail: Kazem.Abhary@unisa.edu.au).

Dr. Romeo Marian is with School of Engineering, University of South Australia, Australia (e-mail: romeo.marian@unisa.edu.au).

24]. However, implementing the hybrid GA is quite complicated while its capability to explore the search space is still limited because the evolutionary process of the obtained solution starts from only one set of initial random solutions.

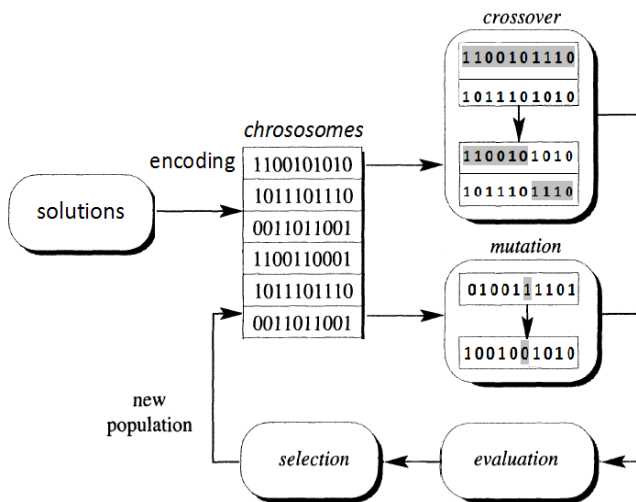


Fig. 1: Traditional structure of GA (adopted from [12])

One simple but effective way to enhance the exploration of *metaheuristics* in general and GA in particular is a procedure called *multi-start* [25, 26]. A basic *multi-start* procedure is simply to run the algorithm multiple times and then pick the best solution among those found over all runs [25, 27]. Certainly, this procedure can help GA improve the probability of jumping out of the local optima and finding the global optimal solution. However, this traditional *multi-start* GA is not very efficient, especially when dealing with large scale optimization problems which usually require a great amount of computing time. A more advanced *multi-start* GA has been developed by Beasley, Bull & Martin [28], in which the GA was sequentially run for a number of times and each run would terminate if either no improvement in average fitness value is made after a certain number of generations or the maximum number of generations is reached. Nevertheless, the effect of elite chromosomes which are guaranteed to pass from one generation to the next was not addressed there yet.

To overcome the above limitations, this paper proposes a so called adaptive restarting GA which is not only capable of restarting the search process if being trapped in local optima but also taking advantage of the elite chromosomes to enhance the probability of finding the global optimal solution. Detail of the proposed adaptive restarting GA will be explained in the next Section.

### III. PROPOSED GENETIC ALGORITHM

This paper mainly focuses on developing an innovative structure for GA. Therefore, the traditional binary chromosome encoding, crossover, mutation and selection as illustrated in Fig. 1 are adopted herein. It should be noted that there have been so many types of chromosome encoding, crossover, mutation as well as selection developed so far in the GA literature, and choosing the most appropriate type is normally a problem-dependent issue. In

contrast, the proposed GA structure as shown in Fig. 2 is a general structure applicable to any problem.

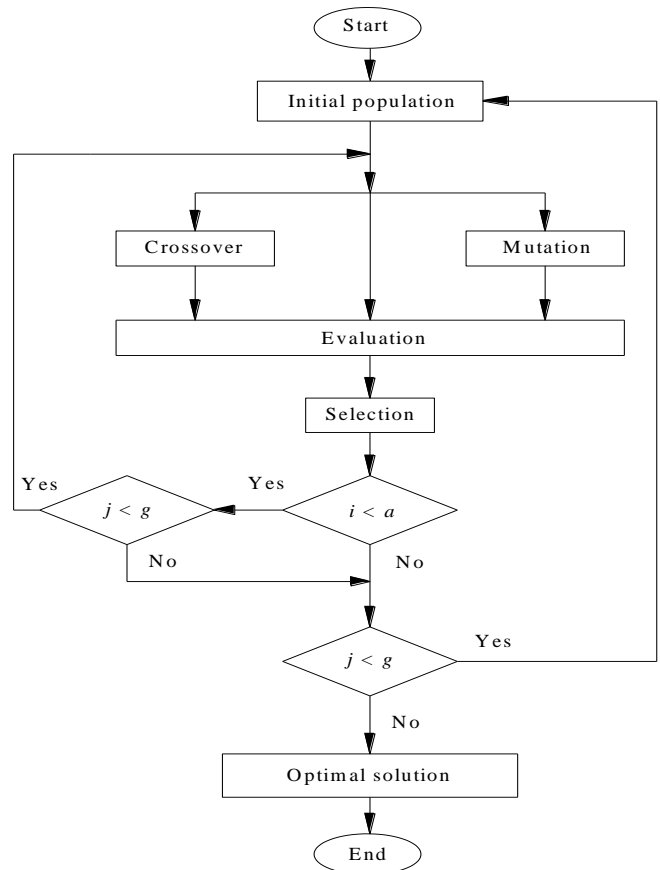


Fig. 2: Structure of the proposed GA

As can be seen from Fig. 2, there are two loops in the proposed adaptive restarting GA. The smaller loop is in essence a traditional GA, which involves genetic operators: crossover, mutation, evaluation and selection. It is noted that the value  $i$  in Fig. 2 represents the current number of generations of the GA in the smaller loop, within which the best solution obtained is not improved; while the value  $j$  is the current generation of the proposed adaptive restarting GA. In the bigger loop, the GA in the smaller loop is restarted with some top quality chromosomes achieved so far. The two loops are controlled by two termination criteria  $a$  and  $g$  which represent the adaptive restarting condition and the maximum number of generations of the proposed GA, respectively.

It should be noted that the traditional *roulette wheel* selection with a small modification is used herein, in which one chromosome in each generation is selected, at most one time, for the next generation to diversify the population. The exploration and exploitation of the proposed adaptive restarting GA can be balanced by appropriate parameters selected by users like population size ( $p$ ), crossover rate ( $c$ ), mutation rate ( $m$ ), adaptive restarting condition ( $r$ ) and elite chromosomes which are guaranteed to pass to the next generation ( $e$ ). It is noted that the parameters  $c$  and  $m$  in this paper are in terms of the number of chromosomes going through the crossover and mutation operations, respectively.

The robustness of the proposed adaptive restarting GA will be demonstrated in the next Section.

IV. PERFORMANCE TESTING

A. Benchmark Functions

Two benchmark functions with different complexities were used to test the performance of the proposed adaptive restarting GA. Details of the two benchmark functions adopted from Nasir and Tokhi [29] are shown in Table 1. Both functions have multiple local optima and proven theoretical global optima as shown in Table 1. The first function (F1) is a two-dimensional function while the second one (F2) is much more complex, with 15 dimensions.

B. Results and Discussions

The proposed adaptive restarting GA was implemented in Matlab to solve the two benchmark functions and its selected parameters are given in Table 2. Performances of the proposed GA as well as five other algorithms are shown in Table 3. It is noted that the performances of four algorithms, namely, Spiral Dynamic Algorithm (SDA), Bacterial Foraging Algorithm (BFA), Hybrid Spiral-Dynamic Bacteria-Chemotaxis algorithm (type R) named HSDBC-R, and Hybrid Spiral-Dynamic Bacteria-Chemotaxis algorithm (type S) named HSDBC-S, for solving the two benchmark functions have been reported in Nasir and Tokhi [29]. Therefore, to make a fair comparison, the maximum number of objective function evaluations of the proposed GA was set exactly the same as in the publication of Nasir and Tokhi [29] as indicated in Table 3. In addition, a traditional GA, which is exactly the same as the proposed GA except two things: (1) no restarting procedure and (2) no guarantee of passing elite chromosome(s) from one generation to the next, was also used herein to verify the effectiveness of the proposed GA. Each algorithm was run for 30 times and their performances in terms of best fitness value (called *Best*), average of fitness values (called *Mean*), and standard deviation of fitness values (called *Std.deviation*) are provided in Table 3.

Table 2: Parameters of the proposed GA

p	c	m	r	e
80	90	6	5	7

As can be seen from Table 3, the first four algorithms, i.e. SDA, BFA, HSDBC-R and HSDBC-S were always stuck in the local optimum when solving the benchmark function (F1); while the traditional GA performed much better, with average fitness value of -0.99 and achieving the global minimum with fitness value of -1.03 most of the times. However, the proposed GA could find the global minimum with success probability of 100%.

Regarding the benchmark function (F2) which is much more complex than the function (F1), no algorithm could find the global minimum, with the total number of objective function evaluations of 80000, as indicated in Table 3. Nevertheless, the traditional GA provided better solutions compared to BFA; while SDA, HSDBC-R and HSDBC-S performed better than the traditional GA. More importantly, the proposed GA outperformed all of the five algorithms in all measures: *Best*, *Mean* as well as *Std.deviation*, as shown in Table 3. The best solution obtained by the proposed GA in 30 runs has fitness value of 1.35, very close to the global minimum of 0.00; while SDA, BFA, HSDBC-R and HSDBC-S achieved the best solutions with fitness values of 22.06, 42.49, 22.03, 14.92 and 36.66, respectively. Regarding the average of fitness values obtained, the proposed GA performed 88.7, 91.1, 85.7, 87.2 and 89.8% better compared to SDA, BFA, HSDBC-R, HSDBC-S and traditional GA, respectively. In terms of consistency, the proposed GA was also much better compared to the five algorithms, indicated by quite small standard deviation of the fitness values achieved in 30 runs, as shown in Table 3.

Based on the results in Table 3, it can be concluded that the proposed GA outperforms SDA, BFA, HSDBC-R, HSDBC-S and traditional GA. For small scale problems like benchmark function (F1), the proposed GA is capable of jumping out of the local optima and always finding the

Table 1: Benchmark functions (adopted from [29])

No.	Name	Dimension	Function	Range	Global minimum
1	Hump (F1)	2	$f_1(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$(-5, 5)^2$	-1.03
2	Rastrigin (F2)	15	$f_2(x) = \sum_{i=1}^{15} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$(-5, 5)^{15}$	0.00

Table 3: Performance comparison

No.	Function	Dimension	Global minimum	No.of.obj.fun. evaluations	Accuracy	SDA [29]	BFA [29]	HSDBC-R [29]	HSDBC-S [29]	Traditional GA	Proposed GA
1	Hump (F1)	2	-1.03	80000	Best	0.00	0.00	0.00	0.00	-1.03	-1.03
					Mean	0.00	0.00	0.00	0.00	-0.99	-1.03
					Std.deviation	0.00	0.00	0.00	0.00	0.05	0.00
2	Rastrigin (F2)	15	0.00	80000	Best	22.06	42.49	22.03	14.92	36.66	1.35
					Mean	56.23	71.77	44.53	49.85	62.31	6.38
					Std.deviation	21.41	10.05	14.00	29.84	14.91	2.42

global optimum. For large scale and complex problems like benchmark function (F2), with 80000 objective-function evaluations, all six algorithms including the proposed GA could not find the global minimum. However, only the proposed GA could achieve solutions which are very close to the global optimal solution. It would be interesting to see how the proposed GA will perform if the number of objective-function evaluations is increased. To answer this question, further investigation of the performance of the proposed GA for solving the benchmark function (F2) is attempted herein; the results are given in Table 4 and Fig. 3. Obviously, the proposed GA is capable of finding the global minimum if the number of objective function evaluations is increased.

Table 4: Global search capability testing

Test function	Dimension	Global minimum	No.of.obj.fun. evaluations	Accuracy	Proposed GA
Rastrigin (F2)	15	0.00	160000	Best	0.16
				Mean	2.48
				Std. deviation	1.42
			320000	Best	0.00
				Mean	0.51
				Std. deviation	0.69
			640000	Best	0.00
				Mean	0.00
				Std. deviation	0.01

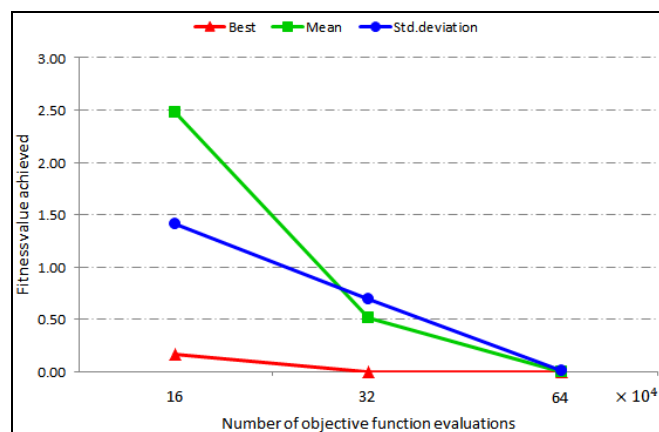


Fig. 3: Visualization of the proposed GA performance (F2)

## V. CONCLUSION

In this paper, a so called adaptive restarting GA has been proposed for the global optimization. With the developed adaptive restarting procedure and elite chromosome strategy, the proposed GA is capable of jumping out of the local optima and finding the global optimum with very high success probability.

The performance of the proposed GA has been tested and compared with five other algorithms, namely, SDA, BFA, HSDBC-R, HSDBC-S as well as traditional GA for solving two global optimization problems. In the first problem, a small scale one, the proposed GA always found the global optimum with 80000 objective-function evaluations, while the five other algorithms could not. In the second problem, a large scale one, no algorithm among the six could achieve the global optimum, with 80000 objective-function evaluations. However, the proposed GA still performed 88.7, 91.1, 85.7, 87.2 and 89.8% better, in terms of average of fitness values obtained in 30 runs, compared to SDA,

BFA, HSDBC-R, HSDBC-S and traditional GA, respectively. In addition, the proposed GA's capability of finding the global solution to this large scale problem, if the number of objective function evaluations is increased, has been confirmed.

In future work, the authors would test and evaluate the robustness of the proposed GA for solving large-scale constrained optimization problems.

## REFERENCES

- [1] Wang, Y., et al., *Two-stage based ensemble optimization framework for large-scale global optimization*. European Journal of Operational Research, 2013. **228**(2): p. 308-320.
- [2] Ng, C.K. and D. Li, *Test problem generator for unconstrained global optimization*. Computers & Operations Research, 2014. **51**(0): p. 338-349.
- [3] Coelho, L.D.S., H.V.H. Ayala, and V.C. Mariani, *A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization*. Applied Mathematics and Computation, 2014. **234**: p. 452-459.
- [4] Hanagandi, V. and M. Nikolaou, *A hybrid approach to global optimization using a clustering algorithm in a genetic search framework*. Computers & Chemical Engineering, 1998. **22**(12): p. 1913-1925.
- [5] Liberti, L. and S. Kucherenko, *Comparison of deterministic and stochastic approaches to global optimization*. International Transactions in Operational Research, 2005. **12**(3): p. 263-285.
- [6] Moles, C.G., P. Mendes, and J.R. Banga, *Parameter estimation in biochemical pathways: a comparison of global optimization methods*. Genome Research, 2003. **13**(11): p. 2467-2474.
- [7] Boender, C.G.E. and H.E. Romeijn, *Stochastic methods*, in *Handbook of Global Optimization*, R. Horst and P.M. Pardalos, Editors. 1995, Kluwer Academic Publishers: Boston
- [8] Victor Paul, P., et al., *Performance analyses over population seeding techniques of the permutation-coded genetic algorithm: An empirical study based on traveling salesman problems*. Applied Soft Computing, 2015. **32**: p. 383-402.
- [9] Fahimnia, B., L. Luong, and R. Marian, *Optimization/simulation modeling of the integrated production-distribution plan: an innovative survey*. WSEAS Transactions on Business and Economics, 2008. **3**(5): p. 52-65.
- [10] Lin, M.H. and J.F. Tsai, *Finding multiple optimal solutions of signomial discrete programming problems with free variables*. Optimization and Engineering, 2011. **12**(3): p. 425-443.
- [11] Holland, J.H., *Adaptation in natural and artificial systems* 1975, Ann Arbor: University of Michigan Press.
- [12] Gen, M. and R. Cheng, *Genetic algorithms and engineering design* 1997, New York John Wiley & Sons.
- [13] Deep, K., et al., *A real coded genetic algorithm for solving integer and mixed integer optimization problems*. Applied Mathematics and Computation, 2009. **212**(2): p. 505-518.
- [14] Dao, S.D. and R. Marian, *Modeling and optimisation of precedence-constrained production sequencing and scheduling for multiple production lines using genetic algorithm*. Computer Technology and Application., 2011. **2**(6): p. 487-499.
- [15] Esen, I. and M.A. Koç, *Optimization of a passive vibration absorber for a barrel using the genetic algorithm*. Expert Systems with Applications, 2015. **42**(2): p. 894-905.
- [16] Tang, P.H. and M.H. Tseng, *Adaptive directed mutation for real-coded genetic algorithms*. Applied Soft Computing, 2013. **13**(1): p. 600-614.
- [17] Michalewicz, Z., *Genetic algorithms + Data structures = Evolution programs* 1996, Berlin Heidelberg: Springer.
- [18] Aiello, G., L.G. Scalia, and M. Enea, *A non dominated ranking multi-objective genetic algorithm and electre method for unequal area facility layout problems*. Expert Systems with Applications, 2013. **40**(12): p. 4812-4819.
- [19] Castelli, M. and L. Vanneschi, *Genetic algorithm with variable neighborhood search for the optimal allocation of goods in shop shelves*. Operations Research Letters, 2014. **42**(5): p. 355-360.
- [20] Boudissa, E. and M. Bounekhla, *Genetic algorithm with dynamic selection based on quadratic ranking applied to induction machine parameters estimation*. Electric Power Components and Systems, 2012. **40**(10): p. 1089-1104.

- [21] Yun, Y., H. Chung, and C. Moon, *Hybrid genetic algorithm approach for precedence-constrained sequencing problem*. Computers & Industrial Engineering, 2013. **65**(1): p. 137-147.
- [22] Shokouhifar, M. and A. Jalali, *An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing*. Expert Systems with Applications, 2015. **42**(3): p. 1189-1201.
- [23] Akpınar, S., G.M. Bayhan, and A. Baykasoglu, *Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks*. Applied Soft Computing, 2013. **13**(1): p. 574-589.
- [24] Mahmoodabadi, M.J., et al., *A novel combination of Particle Swarm Optimization and Genetic Algorithm for Pareto optimal design of a five-degree of freedom vehicle vibration model*. Applied Soft Computing, 2013. **13**(5): p. 2577-2591.
- [25] Martí, R., M.G.C. Resende, and C.C. Ribeiro, *Multi-start methods for combinatorial optimization*. European Journal of Operational Research, 2013. **226**(1): p. 1-8.
- [26] Kessaci, Y., et al., *Parallel evolutionary algorithms for energy aware scheduling*, in *Intelligent Decision Systems in Large-Scale Distributed Environments*, P. Bouvry, H. González-Vélez, and J. Kolodziej, Editors. 2011, Springer Berlin Heidelberg. p. 75-100.
- [27] Dao, S.D., K. Abhary, and R. Marian, *Optimisation of partner selection and collaborative transportation scheduling in Virtual Enterprises using GA*. Expert Systems with Applications, 2014. **41**(15): p. 6701-6717.
- [28] Beasley, D., D.R. Bull, and R.R. Martin, *A sequential niche technique for multimodal function optimization*. Evolutionary Computation, 1993. **1**(2): p. 101-125.
- [29] Nasir, A.N.K. and M.O. Tokhi, *Novel metaheuristic hybrid spiral-dynamic bacteria-chemotaxis algorithms for global optimisation*. Applied Soft Computing, 2015. **27**: p. 357-375.