GLOBAL OPTIMIZATION ACCURACY AND EVOLUTIONARY DYNAMICS OF THE GENERALIZED GENERATION GAP ALGORITHM WITH ADAPTIVE MUTATION

Jason Teo
Center for Artificial Intelligence
School of Engineering and Information Technology
Universiti Malaysia Sabah, Locked Bag No. 2073
88999 Kota Kinabalu, Sabah, Malaysia. Email: jtwteo@ums.edu.my

ABSTRACT

The Generalized Generation Gap (G3) algorithm is one of the most efficient and effective state-of-the-art real-coded genetic algorithms (RCGAs) for unconstrained global optimization. However, its performance on multimodal optimization problems is known to be poor compared to unimodal optimization problems. The G3 algorithm currently relies on crossover operations only. The objective of this paper is to augment the G3 algorithm with adaptive mutation operations which are dynamically activated according to some explicit feedback during the evolutionary optimization process in order to improve its performance for solving multimodal optimization problems. The performance of the enhanced algorithm is compared with its original version based on the global optimization accuracy and the evolutionary dynamics of the optimization process. The proposed algorithm is tested using five benchmark test problems with highly deceptive fitness landscapes. It was found that the performance of the G3 algorithm with adaptive mutation improved significantly in two of the five test problems. In one of these test problems, no optimal solutions could be found previously by the G3 algorithm but can now be solved by the proposed G3 algorithm with augmented adaptive mutation operations.

Keywords: Evolutionary Computation, Global Optimization, Adaptive Mutation, G3 Algorithm, Real-Coded Genetic Algorithms.

1.0 INTRODUCTION

One of the main difficulties currently faced for global optimization problems in continuous search spaces is locating high quality solutions. In other words, solvers for these problems must be able to obtain solutions with a high degree of precision [11]. This is particularly pertinent for continuous multimodal problems where quality rather than computational efficiency is more important as a test of the solver's ability to escape local optima and finding solutions near the global optimum [20]. Moreover, this difficulty is further compounded when the function involves large numbers of variables, which translates into a highly deceptive fitness landscape with very large numbers of local optima [10].

There is currently a strong interest in the application of evolutionary algorithms for solving real-world optimization problems [1]. A large number of recent studies in genetic algorithms (GAs) have focused on the use of real number encoding for solving continuous functions, particularly those with large numbers of variables. In these GAs, the chromosome is a vector of floating point numbers whose length is kept the same as the number of variables to be optimized in the problem, thus directly representing a trial solution to the problem. Such GAs based on real number representations are commonly referred to as real-coded genetic algorithms (RCGAs) [4,8].

One of the most computationally efficient RCGAs is the G3 algorithm developed by Deb et al. [3]. In this algorithm, parent-centric crossover operators are used as the sole mechanism for generating population diversity. The developers of G3 stated that such recombination procedures are sufficient since the procedures are able to generate arbitrary diversity in the offspring population. Hence, the G3 algorithm does not utilize the mutation operator. This algorithm was shown to be highly successful in solving unimodal problems compared to other RCGAs such as Differential Evolution [14], Evolution Strategies [17] and the classical quasi-Newton optimization method [18]. However, it was also reported that the G3 algorithm was much less successful in locating high quality solutions for multimodal problems and the authors highlighted that a new approach which can overcome this shortcoming would be greatly beneficial [3].

The mutation operator in GAs mainly serves to create random diversity in a population of solutions [19]. The G3 algorithm in its current form uses only the crossover operator and appears to suffer from premature convergence due to lack of genetic diversity in large scale multimodal problems. Therefore the objective of this study is to investigate whether the addition of a mutation operator would improve its performance in terms of locating high precision solutions for highly deceptive fitness landscapes. In this set of experiments, an adaptive mutation operator based on the Gaussian distribution is used to introduce random diversity into the offspring population. It is envisaged that the G3 algorithm would be able to escape local optima in these highly deceptive landscapes more effectively with the augmentation of mutational diversity, rather than solely relying on the crossover operator.

This paper is structured as follows. Section 1 presents a brief overview of RCGAs which utilize parent-centric crossover operators. Section 2 outlines the proposed G3AM algorithm. Section 3 presents the five benchmark test problems in unconstrained global optimization and explains the setup of the experiments. Section 4 summarizes the results obtained from this investigation and discusses the performance of the proposed G3AM algorithm against the standard G3 algorithm on the test functions based on the optimization accuracy and evolutionary dynamics. Finally, Section 5 concludes the study and offers some future directions arising from this work.

2.0 PREVIOUS WORK ON THE G3 ALGORITHM

The G3 algorithm was first proposed by Deb et al. in 2002 [3] which was tested on two unimodal and two multimodal problems, with all problems having dimensions of 20 variables. Being a fairly new function optimization algorithm, it has not yet received much attention either in terms of practical application or theoretical analysis. As such, the background literature on the G3 algorithm is similarly limited.

In terms of theoretical analysis, there has been some preliminary work conducted by Ray et al. [16] on the scalability of the G3 algorithm for dimensions of 10, 20 and 50 variables. Similar to the findings of Deb et al. [3], it was observed that the original version of G3 was not well-suited for solving multimodal problems. Further tests on a real-world 44-variable antenna engineering problem further verified this observation. Ray et al. then proposed a new variant of G3 called G3-PCX-II that utilizes a roulette-wheel selection of parents rather than the block selection method used in the original G3. Some improvements were observed in the new version of the G3 algorithm when solving multimodal problems. However, the authors also noted that the improvement in the G3-PCX-II algorithm comes with a trade-off of having a much higher computational cost due to the roulette-wheel selection process.

Some real-world applicative work has also been reported recently using the G3 algorithm in materials science engineering. Mishra and DebRoy [12] used the standard G3 algorithm successfully for locating multiple solutions through a global search of convective heat transfer equations. Rai and DebRoy [15] also used the standard G3 algorithm with success in predicting the optimal values of temperature fields and weld geometry in the keyhole mode laser welding of a particular type of aluminium-magnesium alloy.

Finally, some recent theoretical work was conducted by the original author of the G3 algorithm. Deb proposed a variant of the G3 algorithm that uses a lognormal probability distribution for the main parent component when creating a new offspring rather than a normal distribution [5]. This is to avoid the artificial increase in the probability of creating offspring near the centroid of the multiple parents and to ensure that the new offspring is actually created in the region of the main parent component only. Although no statistical tests were conducted, it was observed that the modified G3 algorithm using the lognormal distribution performed slightly better than the original G3 algorithm for two unimodal and one multimodal optimization problems with dimensions of 20 variables respectively.

3.0 RCGAS AND CROSSOVER OPERATORS

Encoding solutions based on real numbers offers the advantage of defining a large variety of specialized real-coded crossover operators that are able to take advantage of the inherent numerical characteristics. Hence, many different versions of real-coded crossover operators exist for RCGAs. In particular, the blend crossover (BLX) operator [7], the simulated binary crossover (SBX) [2], the unimodal normal distribution crossover (UNDX) operator [13], the simplex crossover (SPX) operator [9] and the parent-centric crossover (PCX) operator [3] have been studied and used extensively. The common theme among these different crossover methodologies is the generation of new offspring which are primarily parent-centered. This essentially defines a probability distribution of offsprings based on some measure of distance among the parents. Further information regarding real-parameter GA recombination operators can be found in [4, 8].

As such, a great majority of RCGAs tend to utilize crossover operators which use some form of arithmetic recombination that in general involves the creation of a new gene *i* for an offspring *z* arising from parents *x* and *y* according to the formula $z_i = x_i + (1-\alpha)y_i$ for some α in [0,1]. Although new genetic material can be created, there is a disadvantage that the range of values is reduced as a result of this averaging process [6]. However, it has been reported in the recent past that RCGAs utilizing some of these parent-based crossover operators exhibit self-adaptive search properties similar to that of evolution strategies and evolutionary programming [4]. Based on these findings, it was argued that depending on the current diversity of the population, these RCGAs self-determine whether exploitation or exploration of the search space will be carried out without requiring an external adaptive control mechanism. Consequently, the use of the mutation operator is foregone is favor of these self-adaptive crossover operators that alone can automatically introduce arbitrary diversity in the offspring population when necessary.

The G3 algorithm is one such parent-based RCGA, which uses the PCX crossover operator without any mutation operator. Although proving highly successful and very efficient for solving continuous unimodal optimization problems, it performed less desirably for highly deceptive fitness landscapes found in large scale multimodal problems with large numbers of local optima [3]. In the next section, the G3 algorithm is proposed to be augmented with an adaptive mutation operator with the hope that its performance in locating solutions with high precision can be improved. This mutation operator is adaptive in that it will only be activated when the G3 algorithm is detected to be prematurely converging to a local optimum. This feedback is obtained via the epsilon metric present within the G3 algorithm. The pseudocode of the proposed algorithm is given below.

3.1 G3 Algorithm with Adaptive mutation (G3AM)

1. From the population $P(t)$ select the best parent and μ-1 other parents randomly.

2. **Crossover:** Generate λ offspring from the chosen parents using the PCX crossover scheme:
   a. Calculate the mean vector $\vec{g}$ of chosen μ parents.
   b. Select one parent $\vec{x}^p$ for each offspring $\vec{y}$ with equal probability.
   c. Calculate the direction vector $\vec{d}^p = \vec{x}^p - \vec{g}$
   d. Calculate for each of the other μ-1 parents the perpendicular distances D_i to the line $\vec{d}^p$.
   e. Calculate the average perpendicular distances $\overline{D}$ from D_i
   f. Create new offspring using:

   $$\vec{y} = \vec{x}^p + w_\varsigma \vec{d}^p + \sum_{i=1, i\neq p}^{\mu} w_\eta \overline{D}\vec{e}^i$$

   where $\vec{e}^i$ are the μ-1 orthonormal bases that span the subspace perpendicular to $\vec{d}^p$, and the parameters ω_ζ and ω_η are zero-mean normally distributed variables with variance σ_ς^2 and σ_η^2.

3. **Mutation:** For each of the λ offspring $\vec{y}$, apply Gaussian mutation $N(0,1)$ with some probability *Uniform*(0,1)<δ to each element in the offspring's chromosome:

   $$\vec{y} \leftarrow \vec{y} + N(0,1)$$

   $$\textbf{\textit{i.f.f.}}\ \vec{d}^p < \varepsilon \ \text{ or } \ \overline{D} < \varepsilon$$

   where ε is a real-numbered metric for detecting premature convergence, which is set to double the required precision of solutions.

4. Choose two parents at random from the population $P(t)$.

5. From the combined sub-population of the chosen two parents and offspring, choose the top two solutions and replace the two chosen parents with these solutions. From the combined sub-population of the chosen two parents and offspring, choose the top two solutions and replace the two chosen parents with these solutions.

4.0 EXPERIMENTAL SETUP

In the experiments conducted, the standard G3 algorithm is compared against the proposed G3AM algorithm. To ensure a fair comparison, all elements are similar except for the addition of the mutation operation in G3AM. The algorithms were run with the following settings as prescribed by the authors of the G3 algorithm [3]:

- The selected parent $\vec{x}^{p}$ for creating offspring is always the current best solution.
- The number of parents μ is set to 3.
- The parameter w_z is set to 0.1.
- The parameter w_h is set to 0.1.
- Solution precision is set to 10^{-20}.

The mutation rate δ is set to 10% as a balance between normal and macro mutation to ensure that sufficient diversity is introduced. The following five continuous multimodal benchmark test functions were selected to compare the performance of G3 with G3AM:

- Rosenbrock's function (F_{Rbk}):

$$f_{Ros} = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

- Rastrigin's function (F_{Ras}):

$$f_{Ras} = 200 + \sum_{i=1}^{n} x_i^2 - 10\cos(2\Pi x_i)$$

- Schwefel's Sine Root function

$$f_{Sch} = 4189829n + \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|})$$

- Griewangk's function (F_{Gri}):

$$f_{Gri} = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$$

- Ackley's Path function (F_{Ack}):

$$f_{Ack} = 20 + e + \exp(-0.2 \sqrt{\frac{1}{n\sum_{i=1}^{n} x_i^2}} - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos 2\Pi x_i)$$

For each of the above test functions, the number of variables n was set to 20 to create a highly deceptive fitness landscape with large numbers of local optima in order to test the algorithms' abilities in escaping early convergence and locating high precision solutions. All of these problems have a minimum of $F^*=0$ when $x_i^*=0$.

To enable a fair comparison, both the G3 and G3AM algorithms were run 10 times each for a maximum of 100,000 functions evaluations. To further ascertain the algorithms' actual abilities in overcoming local optima, a skewed random initialization of variables $x_i \in [-10,-5]$ is used in this study. This type of initialization is important for two reasons [3]: (i) it initializes the population away from the global basin, thereby ensuring that the algorithm must overcome a number of local minima to reach the global basin; and (ii) it prevents the unfair advantage offered by algorithms that generate solutions near the centroid of the parents, since the global basin for test functions generally lie in the center of the prescribed search initialization range.

5.0 RESULTS AND DISCUSSION

This section reports the best solution, the average best solution, the standard deviation and worst solution obtained from the 10 runs of G3 and G3AM on the five benchmark test functions. Table 1 compares the overall best solutions found by both algorithms. As can be seen, the proposed G3AM algorithm dramatically improved on the performance of the standard G3 algorithm for F_{Rbk} and F_{Gri} (marked by *). In F_{Rbk}, G3AM was able to find the global optimal solution in every evolutionary run (100%) whereas the original G3 was only successful 70% of the time. In F_{Gri}, G3AM was able to locate the global optimal solution in two of the runs whereas G3 was not able to locate this global optimum at all. This dramatic improvement was achieved by being able to escape numerous local optima through the mutational diversity offered in moving towards the global basin resulting from the augmentation of the adaptive mutation operations in G3AM.

Table 1. No. of times solution with required precision of 10^{-20} was found over 10 evolutionary runs.

Test Function	G3 Algorithm	G3AM Algorithm
F_{Rbk}	7	10*
F_{Ras}	0	0
F_{Sch}	0	0
F_{Gri}	0	2*
F_{Ack}	0	0

Table 2 compares the overall best solutions found by both algorithms. Entries denoted as <1.0e-20 indicate that the global optimum was found by the respective algorithms. As shown in the table, the G3AM algorithm was able to find slightly superior solutions than the G3 algorithm for the two test problems in which the global optimum was not found, that is for F_{Ras} and F_{Ack} but G3 found a marginally better solution for F_{Sch}.

Table 2. Overall best solution obtained over 10 evolutionary runs.

Test Function	G3 Algorithm	G3AM Algorithm
F_{Rbk}	<1.0e-20	<1.0e-20
F_{Ras}	5.9000e+02	4.1290e+02
F_{Sch}	7.9584e+03	8.6493e+03
F_{Gri}	9.9920e-16	<1.0e-20
F_{Ack}	1.4445e+01	1.4216e+01

Again, the average of the best solutions found for G3AM clearly outperformed the standard G3 algorithm as shown in Table 3. The average best solutions for F_{Rbk} was far superior in G3AM compared to G3 due to the fact that the best solutions in each of the runs in the former was able to locate itself within the global basin of the required solution precision of 10^{-20}. Even for the test functions which could not find the global optimum, G3AM still produced slightly better solutions on the average compared to G3, as can be seen from the results of F_{Ras} and F_{Ack}. However, G3 had a slightly better average best solution in F_{Sch}.

Table 3. Average of the best solution obtained and standard deviation over 10 evolutionary runs.

Test Function	*G3 Algorithm*	G3AM Algorithm
F_{Rbk}	2.2978e-06 ±7.2631e-06	9.9035e-21 ± 1.0409e-22
F_{Ras}	8.3770e+02 ± 9.6130e+01	6.0224e+02 ± 1.1236e+02
F_{Sch}	7.9980e+03 ± 2.3944e+01	8.7580+e03 ± 6.4316e+01
F_{Gri}	1.6709e-02 ± 3.0820e-02	1.6709e-02 ± 3.0820e-02
F_{Ack}	1.4766e+01 ± 2.7784e-01	1.4652e+01 ± 3.2261e-01

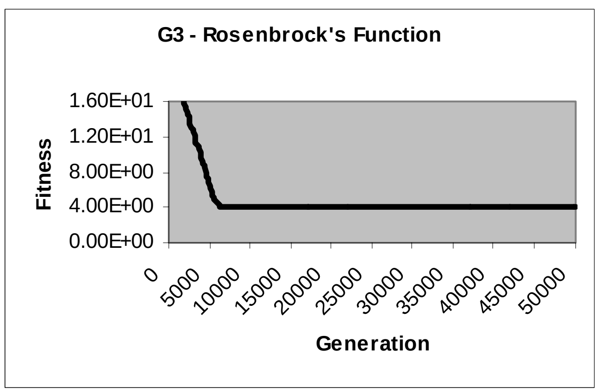
As clearly demonstrated by the results shown in Table 4, the worst of the best solutions found by G3AM over 10 evolutionary runs was still superior than G3 for all test functions except for F_{Gri} where both had the result and F_{Sch} where G3 was slightly superior. Comparing the worst of the solutions found for F_{Rbk}, the modified G3AM algorithm was able to outperform G3 by more than 14 orders of magnitude. This evidence supports the usefulness of the adaptive mutational diversity introduced in G3AM that allows the algorithm to still locate very high quality solutions even in the worst performing runs.

Table 4. Worst of the best solutions obtained over 10 evolutionary runs.

Test Function	G3 Algorithm	G3AM Algorithm
F_{Rbk}	2.2969e-06	<1.0e-20
F_{Ras}	9.2728e+02	8.0590e+02
F_{Sch}	8.0389e+03	8.8671e+03
F_{Gri}	1.0062e-01	1.0062e-01
F_{Ack}	1.5355e+01	1.5264e+01

5.1 Evolutionary Dynamics: G3 vs. G3AM

In this next section, we analyze the evolutionary dynamics of both G3 and G3AM in order to ascertain the convergence properties of these algorithms over the evolutionary search process. The y-axis of the graphs by necessity are of different ranges due to the significantly different solution precision and evolutionary dynamics of the algorithms.



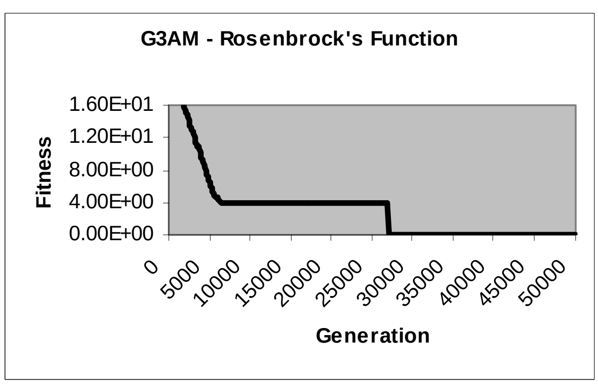


Fig 1. G3 (top) vs. G3AM (bottom): Evolutionary dynamics for Rosenbrock's Function.

Fig. 1 and Fig. 2 depict the typical runs during the evolutionary optimization process of G3 and G3AM in which the global optimal solution cannot be found by G3 but was able to be successfully located by G3AM. From these graphs, it can be clearly seen that the G3 algorithm lost genetic diversity extremely quickly in both of these test functions. Premature convergence can be seen to occur as early as within the first 5,000 and 2,000 out of the permissible 50,000 generations in the original G3 algorithm (equivalent to 100,000 function evaluations as 2 new offspring are created every generation) where the fitness of the best solution was still well beyond the value of 3 for

F_{Rbk} and 10e-16 for F_{Gri} respectively. The converse is true for the proposed G3AM where it only began to converge after locating the global basin near the optimal value of 0.0. Although not visible in the graphs due to the precision acquired by G3AM, continuous improvement in the best solution occurred up to generation 33,000 and 10,500 respectively for F_{Rbk} and F_{Gri}.

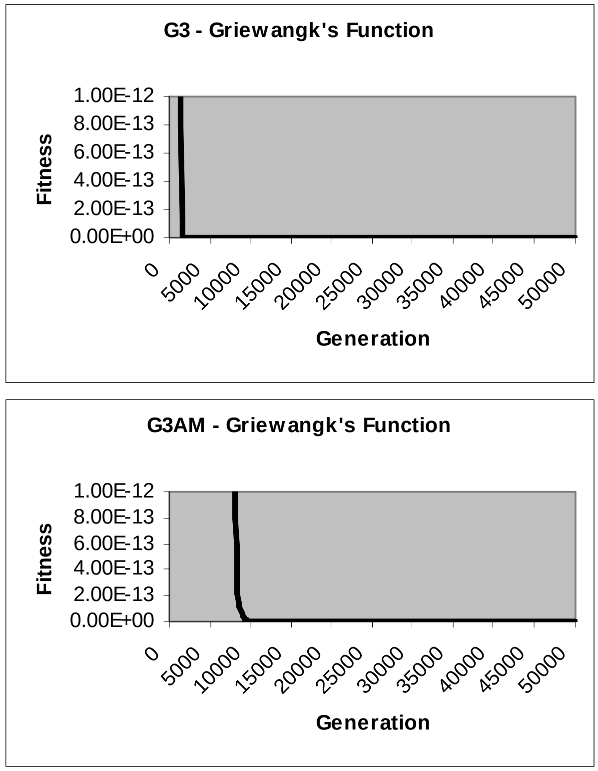


Fig 2. G3 (top) vs. G3AM (bottom): Evolutionary dynamics for Griewangk's Function.

5.2 Advantages and Disadvantages of the G3AM Algorithm

As has been clearly demonstrated, the main advantage provided through the augmentation of G3 with adaptive mutation is that the performance of the G3 algorithm using PCX crossover has improved significantly for solving multimodal function optimization problems. Another advantage is that since the added mutation operator works adaptively, the genetic operation will be carried out automatically by the new G3 algorithm. However, this adaptiveness depends on the G3 algorithm receiving some explicit feedback from the evolutionary optimization process, which comes in the form of the epsilon metric. This would be the only disadvantage of the new G3 algorithm since it now requires the calculation of an extra metric. However, since epsilon is calculated only once for $\vec{d}^{\,p}$ and once for $\overline{D}$, it has minimal effect on the overall run-time complexity of the G3 algorithm.

6.0 CONCLUSION AND FUTURE WORK

This investigation has empirically shown G3 real-coded genetic algorithm can benefit greatly from the augmentation of adaptive mutation to its genetic operations in promoting solution diversity in the population and avoiding premature convergence. The adaptive mutational diversity introduced in the proposed G3AM algorithm showed highly competitive results against the standard G3 algorithm on five benchmark continuous multimodal test functions. The G3AM algorithm dramatically outperformed the G3 algorithm in two of the test functions in terms of solution accuracy. This paper has demonstrated that an adaptive Gaussian-based mutation can significantly improve a real-coded genetic algorithm's ability to escape local optima in a highly deceptive fitness landscape, thereby enabling the search to locate global optimal solutions. For future work, it would be worthwhile to extend this approach to other well-known RCGAs and even more general evolutionary algorithms such as Differential

Evolution (DE) that solely rely on the crossover operation for generating genetic diversity. Also, it would be interesting to investigate whether a self-adaptive, rather than adaptive approach for the mutational diversity operations, can further enhance the search quality of the G3 algorithm.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the support of the Ministry of Higher Education, Malaysia, for providing funding under the Fundamental Research Grant Scheme (FRGS).

REFERENCES

[1] T. Back, D.B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. Institute of Physics Publishing, Bristol, Philadelphia, 1997.

[2] K. Deb. *Optimization methods for Engineering Design*. Prentice-Hall, New Delhi, India, 1995.

[3] K. Deb, A. Anand, and D. Joshi. “A computationally efficient evolutionary algorithm for real-parameter evolution”. *Evolutionary Computation*, Vol. 10 No. 4, 2002, pp. 371–395.

[4] K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chicester, UK, 2001.

[5] K. Deb. “A population-based algorithm-generator for real-parameter optimization”. *Soft Computing*, Vol. 9, 2005, pp. 236-253.

[6] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin, 2003.

[7] J.L. Eshelman and J.D. Schaffer. “Real-coded genetic algorithms and interval-schemata”. In L.D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pp. 187–202. Morgan Kaufmann, San Mateo, California, 1993.

[8] F. Herrera, M. Lozano, and J.L. Verdegay. “Tackling real-coded genetic algorithm: operators and tools for the behavioral analysis”. *Artificial Intelligence Reviews*, Vol 12 No. 4, 1998, pp.265–319.

[9] T. Higuchi, S. Tsutsui, and M. Yamamura. “Theoretical analysis of simplex crossover for real-coded genetic algorithms”. In M. Schoenauer et al., editor, *Parallel Problem Solving from Nature (PPSN-VI)*. Springer, Berlin, 2000.

[10] S.Y. Ho, L.S. Shu, and J.H. Chen. “Intelligent evolutionary algorithms for large parameter optimization problems”. *IEEE Transactions on Evolutionary Computation*, Vol. 8 No. 6, 2004, pp.522–541.

[11] H. Kita. “A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms”. *Evolutionary Computation*, Vol. 9 No.2, 2001, pp.223–241.

[12] S. Mishra and T. DebRoy. “A computational procedure for finding multiple solutions of convective heattransfer equations”. *Journal of Physics D: Applied Physics*, Vol. 28, 2005, pp.2977-2985.

[13] I. Ono and S. Kobayashi. “A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover”. In T. Back, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-7)*. Morgan Kaufmann, San Francisco, 1997.

[14] K.V. Price. “An introduction to differential evolution”. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pp. 79–108. McGraw-Hill, UK, 1999.

[15] R. Rai and T. DebRoy. “Tailoring weld geometry during keyhole mode laser welding using a genetic algorithm and a heat transfer model”. *Journal of Physics D: Applied Physics*, Vol. 39, 2006, pp.1257-1266.

[16] T. Ray, N. Venkataralayu, K.S. Won and K.P. Chan. "Study on the Behaviour and Implementation of Parent Centric Crossover within the Generalized Generation Gap Model" in *Proceedings of the 2004 Congress on Evolutionary Computation*, 2004, pp.1996-2003.

[17] I. Rechenberg. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, Germany, 1973.

[18] G.V. Reklaitis, A. Ravindran, and K.M. Ragsdell. *Engineering Optimization Methods and Applications*. John Wiley and Sons, New York, 1983.

[19] W.M. Spears. "Crossover or mutation?" In L.D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pp. 221–238. Morgan Kaufmann, San Mateo, California, 1993.

[20] Z. Tu and Y. Lu. "A robust stochastic genetic algorithm (StGA) for global numerical optimization". *IEEE Transactions on Evolutionary Computation*, Vol. 8 No.5, 2004, pp.459–470.

BIOGRAPHY

Jason Teo is a lecturer in computer science at the School of Engineering and Information Technology, Universiti Malaysia Sabah. He received his doctorate in information technology from the University of New South Wales, Australia, researching Pareto artificial evolution of virtual legged organisms. He has over 50 publications in the areas of evolutionary computation, artificial life, evolutionary robotics and swarm intelligence. His current research interests include multi-objective robotics, auto-constructive artificial life, co-evolutionary game AI and evolutionary optimization. He is a professional member of the ACM and IEEE and also serves on the review board of a number of international journals.