

# Crossover and Mutation Operators of Genetic Algorithms

Siew Mooi Lim, Abu Bakar Md. Sultan, Md. Nasir Sulaiman, Aida Mustapha, and K. Y. Leong

**Abstract**—Genetic algorithms (GA) are stimulated by population genetics and evolution at the population level where crossover and mutation comes from random variables. The problems of slow and premature convergence to suboptimal solution remain an existing struggle that GA is facing. Due to lower diversity in a population, it becomes challenging to locally exploit the solutions. In order to resolve these issues, the focus is now on reaching equilibrium between the explorative and exploitative features of GA. Therefore, the search process can be prompted to produce suitable GA solutions. This paper begins with an introduction, Section 2 describes the GA exploration and exploitation strategies to locate the optimum solutions. Section 3 and 4 present the lists of some prevalent mutation and crossover operators. This paper concludes that the key issue in developing a GA is to deliver a balance between explorative and exploitative features that complies with the combination of operators in order to produce exceptional performance as a GA as a whole.

**Index Terms**—Crossover operator, mutation operator, exploitation, exploration.

## I. INTRODUCTION

The main search operator in Genetic algorithms (GA) is the crossover operator which equally as significant as mutation, selection and coding in GA. The crossover operator functions primarily in the survey of information that is accessible through the search space, which inadvertently improves the behavior of the GA. On another note, mutation is a secondary operator. It functions to alter the genes of the offspring. A mutator will diversify the existing population and this inadvertently allows GAs to exploit promising areas of the search space thus avoiding local solutions [1]. Some of the mutation operators are designed to explicitly overcome certain types of issues over others [2]. The performance among all the comparative of GA operators are easily validated and compared through unbiased test problems from the literature, which are diverse in properties in terms of complexity and modality. This study substantially contributes in reviewing some prevalent mutation and crossover operators. The operators maintain a good balance between explorative and exploitative strategies while manufacturing the optimum GA solutions.

## II. ACHIEVING EXPLORATION AND EXPLOITATION IN GENETIC ALGORITHM

A crossover or mutation can function as an exploration or exploitation operator [3], [4]. Although optimization algorithms with higher degree of exploitation may have

higher convergence speed, the challenge lies in locating the optimal solution and chances are it may not get past a local optimum. On the other hand, algorithms that favor exploration over exploitation might consume more time in locating the global optimum, that is, coincidentally, due to its less sophisticated candidate solutions. A comprehensive survey in exploration and exploitation in evolutionary algorithms is reported [5].

Ref. [6], [7] reported that real parameter crossover operators is equipped with self-adaptive features which enable them to produce different degrees of exploration or exploitation by looking at how they deal with the current diversity of the population. The exploration process is commenced by generating additional diversity starting from the current one. This means finding new points in the search space, which has not been investigated before. Another alternative is for the crossover to start the exploitation process by using the earlier generated diversity for creating better elements; which improves and combines the traits of the currently known solution(s). Elsayed *et al* [8] has proposed a new GA which consists of three-parent crossover that produces three new offspring. One of these offspring is used for exploration process while the other two offspring are to promote exploitation. Intensifications and diversifications are synonymous to exploitation and exploration, which have been introduced in Glover's Tabu search [9].

To add to that, Herrera *et al* [10] also shed some light on the basic understanding of the availability of a crossover adapting to different degrees of exploration and exploitation. To illustrate this concept, let's consider two genes,  $c_a, c_b \in [x, y]$  to be combined with  $\alpha_i = \min \{c_a, c_b\}$  and  $\beta_i = \max \{c_a, c_b\}$ . The action interval  $[x, y]$  of these genes is further divided into three intervals:  $[x, \alpha_i]$ ,  $[\alpha_i, \beta_i]$  and  $[\beta_i, y]$  in order to classify the exploration and exploitation zones.

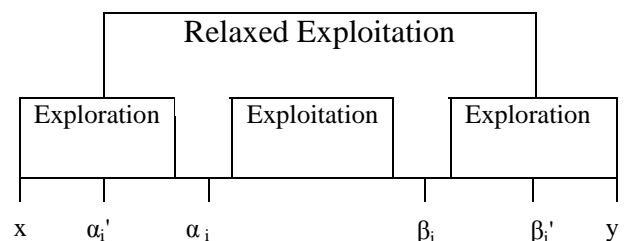


Fig. 1. Action interval for two genes (Ca and Cb) [10].

As shown in Fig. 1, the interval with both genes being the extremes is declared as an exploitation zone because when a crossover produces any genes,  $z_i$  in this interval, it fulfills the properties of  $\max \{|z_i - \alpha_i|, |z_i - \beta_i|\} \leq |\alpha_i - \beta_i|$ . On the other hand, the two intervals that remain on both sides are exploration zones, because the said property is not fulfilled.

The zone with extremes  $\alpha_i'$  and  $\beta_i'$  is declared as relaxed exploitation zone. This region  $[\alpha_i', \beta_i']$  has the properties of  $\alpha_i' \leq \alpha_i$  and  $\beta_i' \geq \beta_i$ . Therefore, based on the genes generated on these intervals, any crossover operators for RCGAs can determine their exploration and exploitation degrees.

### III. MUTATION OPERATORS

Studies have been carried out on the varieties of mutation techniques to improve the GAs performance over the years [11]. The purpose of mutation operation is to change the genes of the offspring and to increase the diversity of the population. This process enables GAs to jump out of local or suboptimal solutions to avoid premature convergence. The following paragraph presents some prevalent mutation operators [12]:

*Mirror mutation* and *binary bit-flipping mutation* [13], [14] are alike in that the *mirror mutator* replaces a gene with its mirror value at the middle point of the boundary interval for the gene, whereas in bit-string representation GA, *bit-flip mutation* remains unchanged. Order based GAs and grouping GAs are instances of GA minus the *bit-flip mutation*. *Random (uniform) mutation* [15] is a common mutation operator based on Gaussian distribution whereby the user specifies a range of uniform random value to replace the value of the chosen gene.

*Mutation based on directed variation techniques* [16] make use of the feedback information from the current population to make changes to certain individuals. The direction of *mutation based on co-evolutionary technique* [17] is determined by a solution vector.

*Directed mutation* [18] is based on gradient or extrapolation. The *directed mutation* deterministically finds a new point in the population using the information applied in the previous generations. *Directed mutation based on momentum* [19] is a standard *Gaussian mutation*, which is used to speed up the gradient descent training of neural networks. The existing momentum functions as a mutator for each component of an individual.

*Covariance matrix adaptation evolution strategy (CMA-ES)* [20] was recommended by experts as an outstanding parametric optimization algorithm. *Muhlenbein's mutation (MM)* [21] generates offspring with alleles and *logarithmic mutation (LM)* [22] alters a randomly chosen allele.

The wavelet theorem gave rise to *wavelet mutation* [23]. This mutator performed the best among other RCGAs when applied to the economic load dispatch and tuning an associative-memory neural network. *Edge mutation* and *tension vector mutation* [24] are based on the breadth-first (BF) force-based and tension vector methods respectively.

*Power mutation (PM)* [25] and *polynomial mutation (PLM)* [26] are based on power distribution and polynomial distribution respectively. The strength of *PM* is regulated by its index, whereby small (large) index value produces small (large) diversity. *Makinen, Periaux and Toivanen mutation (MPTM)* [27] has been proved to solve constrained and multidisciplinary shape optimization problems. *PM* performs better than *MPTM* and *Non-Uniform Mutation (NUM)* when all three mutators were combined with *Laplace Crossover (LX)*. Of all of them, *PLM* is one of the most widely used operator as it has been applied in single

and multi-objective optimization problems [28], [29].

The *breeder GA mutation (BGA)* [30] produces solutions on equal sides of a chromosome. The perturbation created in a chromosome is within a 10% range of what is permissible for a particular gene. The design concepts of *discrete and continuous modal mutations* [31] are similar to that of the *BGA*.

*Non-uniform mutation (NUM)* [32] possesses a fine-tuning capability whereby its action depends on the number of the population in order to reach equilibrium between exploration and exploitation. The search was performed uniformly at the beginning and very locally towards the end of the search. Such an approach is categorized as uni-process driven. *Multi-non-uniform mutation (MNUM)* [33] escalates the genetic diversity of a candidate individual. The *MNUM* can perform a uniform search with local fine-tuning thus increasing the exploitative capability. *Principal component analysis mutation (PCA)* [34] on the other hand, is applied on the IIR Filter design problem. It produces higher level of diversity in the population as compared to *uniform* and *non-uniform mutation*.

*Adaptive mutation* [35] operator uses the simulation of gradient or counter-gradient direction in its searching strategies. It relies on the frequency of the best chromosomes's genes and it possesses an adaptive feature. On the contrary, *adaptive directed mutation (ADM)* [36] incorporates the strategies of local directional search and the adaptive random search to avoid the concentration of each chromosome caused by a crossover operator. This strategy can also prevent an unsystematic search of the system due to random mutation.

*Covariance matrix adaptation evolution strategy (CMA-ES)* [37] was recommended by experts as an outstanding parametric optimization algorithm. *Muhlenbein's mutation (MM)* [38] generates offspring with alleles and *logarithmic mutation (LM)* [39] alters a randomly chosen allele.

### IV. CROSSOVER OPERATORS

Two main approaches of crossover development are parent centric and mean centric operators. The parent centric approach generates offspring near each of the parents whereas mean centric generates offspring solutions near the centroid of the parents, which is in the vicinity of mean of the participating parents. Some well-known crossover operators are reflected in the following paragraph [12]. The discussions start with the classical examples.

*Single point crossover*, [40] detects one crossover point at random before splitting parents at this crossover point thereby producing offspring by exchanging tails. The common crossover probability is within the range of 0.2 - 1.0. *n-point crossover* [41] is a generalization of the single point crossover. The *n* crossover points are randomly identified, after which they split along those points and convene, and subsequently alternating between parents.

*Uniform crossover* assigns 'heads' to one parent, 'tails' to the other. A coin is flipped for each gene of the first child and an inverse copy of the gene is created for the second child. Inheritance does not rely on position.

In an experiment conducted by Lobo [42], it was reported that  $P_s$  has an impact on the exploration process. In this case,

the  $P_s$  is proportional to the search space. Contrary to that, *uniform crossover* outperforms *two-point crossover* even with a smaller selection of population in a particular problem.

*Discrete crossover* [30] is analogous to the classical *one-point* and *uniform crossover*. *Uniform crossover* is the enhancement of *UNDX* [43]. *Single point*, *n-point* and *uniform crossover* operators have been employed in binary and real-coded GAs.

*Arithmetical crossover* [44] involves arithmetic procedures aimed at generating new offspring, which are sandwiched between the parents. A whole arithmetic crossover has been applied in an adaptive pair bond GA [45]. On the other hand, *Flat crossover (BLX-0.0)* [46] utilizes uniform distribution to randomly produce offspring between the genes of the parents.

The above mentioned operators were among the few initial attempts which implemented in an exploitative search. They generate offspring only in the region bounded by the parents thus causing premature convergence. However, this problem was overcome by the other crossover operators whereby the offspring is produced in the exploration region near the parents, and not only within the region bounded by them.

*Simulated binary crossover (SBX)* [47] is devised to simulate the effect of *one-point crossover* on a string of binary alphabets in a continuous domain. The Taguchi method is incorporated to improve the robustness and degree of exploitation of the algorithm. A SBX based multilevel thresholding real coded genetic algorithm has been applied to segment the medical brain images into normal and abnormal tissues.

*Wright's heuristic crossover* [48] employs the fitness function value of the parent solutions to produce a child solution from a pair of parents with a bias towards the better one. *Unimodal normal distribution crossover (UNDX)* [43], *Parent centric crossover (PCX)* [49] and *UNDX-m* [50] are crossover operators with several parents.

*UNDX* generates offspring using a normal distribution defined by three parents. Two or more offspring solutions are created around the center of the mass. Both of the operators are effectively used with an elite-preserving, steady-state, scalable, and computationally fast evolutionary model such as the G3 model. Additionally, *UNDX-m* is a multi-parental extension of the *UNDX* and it possesses a better search ability of the *UNDX*. It properly preserves the statistics of mean vector and the covariance matrix of the population.

*Dynamic fuzzy connective based crossover* [51] strive to strike a balance between exploration and exploitation during the evolution process. *Heuristic fuzzy connective based crossover* works on retaining the diversity and speeds up the convergence by exploring the region near to the superlative parents.

Being a multi-parent crossover operator, the *simplex crossover (SPX)* [52] generates offspring vector values by uniformly sampling values from simplex formed by  $k$  ( $2 \leq k \leq$  number of parameters + 1) parent vectors.

In a linear crossover, two parents generate three offspring. Through an offspring selection mechanism, it will determine the two most promising offspring among the three to

substitute their parents in the population.

*Average bound crossover* [23] consists of both average and bound crossovers. In *average bound crossover*, a pair of parents produces four offspring solutions from which two of the best offspring are selected to substitute the parents. *Hybrid crossover* [53] use various types of crossovers, a number of offspring solutions from the same set of parents are formed.

It is claimed that representation of individuals strongly influences the explorative and exploitative strengths of GA. However, *Geometrical crossover* [54] does not rely on representation and it is defined based on the distance of the solution space.

Examples of parent-centric approaches include *blend crossover*, *fuzzy min-max*, *fuzzy recombination*, *SBX* etc. On the other hand, examples of mean-centric approaches are *SPX*, *UNDX* and so on.

## V. SUMMARY

The key to a successful implementation of GAs primarily depends on the efficient crossover and mutation search operators to guide the system toward global optima i.e. locating the global minimum or maximum of the objective functions to these problems. A balance between the two explorative and exploitative features of GA, which are characterized by the two operators, is imperative in order to speed up the search process to produce quality GA solutions.

## REFERENCES

- [1] I. Korejo, S. Yang, and C. Li, "A directed mutation operator for real coded genetic algorithms," *Applications of Evolutionary Computation*, Springer, pp. 491-500, 2010.
- [2] W. Gong, Z. Cai, and D. Liang, "Adaptive ranking mutation operator based differential evolution for constrained optimization," *IEEE Transactions, Cybernetics*, vol. 45, no. 4, pp. 716-727, 2015.
- [3] N. Fang, J. Zhou, R. Zhang, Y. Liu, and Y. Zhang, "A hybrid of real coded genetic algorithm and artificial fish swarm algorithm for short-term optimal hydrothermal scheduling," *International Journal of Electrical Power and Energy Systems*, vol. 62, pp. 617-629, 2014.
- [4] S. Kumar, V. K. Sharma, and R. Kumari, "A novel hybrid crossover based artificial bee colony algorithm for optimization problem," *Arxiv Preprint Arxiv*, 2014.
- [5] M. Črepinšek, S. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, pp. 35, 2013.
- [6] H. Kita, "A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms," *Evolutionary Computation*, vol. 9, no. 2, pp. 223-241, 2001.
- [7] H. Beyer and K. Deb, "On self-adaptive features in real-parameter evolutionary algorithms," *Evolutionary Computation, IEEE Transactions*, vol. 5, no. 3, pp. 250-270, 2001.
- [8] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 57-69, 2014.
- [9] F. Glover and M. Laguna, *Tabu Search*, Springer, 2013.
- [10] F. Herrera, M. Lozano, and A. M. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *International Journal of Intelligent Systems*, vol. 18, no. 3, pp. 309-338, 2003.
- [11] P. Tang and M. Tseng, "Adaptive directed mutation for real-coded genetic algorithms," *Applied Soft Computing*, 2012.
- [12] S. M. Lim, "Crossover and mutation operators of real coded genetic algorithms for global optimization problems," unpublished Ph.D thesis, University Putra Malaysia, Malaysia, 2016.
- [13] L. Davis, *Handbook of Genetic Algorithms Van Nostrand Reinhold New York*, 1991.

- [14] E. Falkenauer and A. Delchambre, "A genetic algorithm for bin packing and line balancing," in *Proc. IEEE International Conference on Robotics and Automation*, 1992, pp. 1186-1192.
- [15] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," New York: Addison-Wesley, 1989.
- [16] Q. Zhou and Y. Li, "Directed variation in evolution strategies," in *Proc. IEEE Transactions on Evolutionary Computation*, 2003, vol. 7, no. 4, pp. 356-366.
- [17] A. Berry and P. Vamplew, "Pod can mutate: A simple dynamic directed mutation approach for genetic algorithms," 2004.
- [18] D. Bhandari, N. R. Pal, and S. K. Pal, "Directed mutation in genetic algorithms," *Information Sciences*, vol. 79, no. 3, pp. 251-270, 1994.
- [19] L. Temby, P. Vamplew, and A. Berry, "Accelerating real-valued genetic algorithms using mutation-with-momentum," *AI 2005: Advances in artificial intelligence*, Springer, pp. 1108-1111, 2005.
- [20] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159-195, 2001.
- [21] H. Mühlhain, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," *Parallel Computing*, vol. 17, no. 6, pp. 619-632, 1991.
- [22] P. Pongcharoen, W. Chainate, and P. Thapatsuwat, "Exploration of genetic parameters and operators through travelling salesman problem," *Science Asia*, vol. 33, no. 2, pp. 215-222, 2007.
- [23] S. Ling and F. F. Leung, "An improved genetic algorithm with average-bound crossover and wavelet mutation operations," *Soft Computing*, vol. 11, no. 1, pp. 7-31, 2007.
- [24] D. Vrajitoru and J. DeBoni, "Hybrid real-coded mutation for genetic algorithms applied to graph layouts," in *Proc. 2005 Conference on Genetic and Evolutionary Computation*, pp. 1563-1564, 2005.
- [25] K. Deep and M. Thakur, "A new mutation operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 211-230, 2007b.
- [26] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30-45, 1996.
- [27] Toivanen, Raino, J. Périaux, and F. Cloud Cedex, "Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms," *Int. J. Numer. Meth. Fluids*, vol. 30, pp. 149-159, 1999.
- [28] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," 1994.
- [29] K. Deb, "Multi-objective optimization using evolutionary algorithms," *John Wiley & Sons*, 2001.
- [30] D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm," *Evolutionary Computation*, vol. 1, no. 1, pp. 25-49, 1993.
- [31] H. Voigt and T. Anheyer, "Modal mutations in evolutionary algorithms," in *Proc. IEEE World Congress on Computational Intelligence*, 1994, pp. 88-92.
- [32] Z. Xinchao, "Simulated annealing algorithm with adaptive neighborhood," *Applied Soft Computing*, vol. 11, no. 2, pp. 1827-1836, 2011.
- [33] C. R. Houck, J. Joines, and M. G. Kay, "A genetic algorithm for function optimization: A matlab implementation," *NCSU-IE TR*, vol. 95, no. 9, 1995.
- [34] C. Munteanu, and V. Lazarescu, "Improving mutation capabilities in a real-coded genetic algorithm: Evolutionary image analysis, signal processing and telecommunications," Springer, pp. 138-149, 1999.
- [35] M. Chen and F. Liao, "Adaptive mutation operators and its applications," *Journal of Dayeh University*, vol. 7, no. 1, pp. 91-101, 1998.
- [36] P. Tang and M. Tseng, "Adaptive directed mutation for real-coded genetic algorithms," *Applied Soft Computing*, vol. 13, no. 1, pp. 600-614, 2013.
- [37] N. Hansen and A. Ostermeier, "Completely de-randomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159-195, 2001.
- [38] H. Mühlhain, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," *Parallel Computing*, vol. 17, no. 6, pp. 619-632, 1991.
- [39] P. Pongcharoen, W. Chainate, and P. Thapatsuwat, "Exploration of genetic parameters and operators through travelling salesman problem," *Science Asia*, vol. 33, no. 2, pp. 215-222, 2007.
- [40] J. H. Holland, "Adaption in natural and artificial systems," University of Michigan press, 1975.
- [41] L. J. Eshelman, "Crossover operator biases: Exploiting the population distribution," in *Proc. the Seventh International Conference on Genetic Algorithms*, pp. 354-361, 1997.
- [42] F. G. Lobo and D. E. Goldberg, "The parameter-less genetic algorithm in practice," *Information Sciences*, vol. 167, no. 1, pp. 217-232, 2004.
- [43] I. Ono, H. Satoh, and S. Kobayashi, "A real-coded genetic algorithm for function optimization using the unimodal normal distribution crossover," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 14, pp. 1146-1155, 1999.
- [44] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Springer, 1996.
- [45] T. Y. Lim, M. A. Al-Betar, and A. T. Khader, "Adaptive pair bonds in genetic algorithm: An application to real-parameter optimization," *Applied Mathematics and Computation*, vol. 252, pp. 503-519, 2015.
- [46] N. J. Radcliffe, "Equivalence class analysis of genetic algorithms," *Complex Systems*, vol. 5, no. 2, pp. 183-205, 1991.
- [47] R. Agrawal and B. Deb, "Simulated binary crossover for continuous search space," 1994.
- [48] A. H. Wright, "Genetic algorithms for real parameter optimization," in *Foundations of Genetic Algorithms I, FOGA*, G. J. E. Rawlins. Ed. pp. 205-218, 1990.
- [49] K. Deb, A. Anand, and D. A. Joshi, "Computationally efficient evolutionary algorithm for real-parameter optimization," *Evolutionary Computation*, vol. 10, no. 4, pp. 371-395, 2002.
- [50] H. Kita, I. Ono, and S. Kobayashi, "Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms," *Evolutionary Computation, CEC 99*, 1999.
- [51] F. Herrera and M. Lozano, "Adaptation of genetic algorithm parameters based on fuzzy logic controllers," *Genetic Algorithms and Soft Computing*, vol. 8, pp.95-125, 1996.
- [52] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent recombination with simplex crossover in real coded genetic algorithms," in *Proc. the Genetic and Evolutionary Computation Conference*, vol. 1, pp. 657-664, 1999.
- [53] R. Ramli, A. R. Rosshairy, "Average concept of crossover operator in real coded genetic algorithm," 2005.
- [54] A. Moraglio, Y. Kim, Y. Yoon, and B. Moon, "Geometric crossovers for multiway graph partitioning," *Evolutionary Computation*, vol. 15, no. 4, pp. 445-474, 2007.

**Siew Mooi Lim** is an assistant professor in Faculty of Computer Science at University Malaysia of Computer Science and Engineering. She is also the coordinator of Postgraduate Studies and Research at the same university. She completed her Ph.D in intelligent computing at University Putra Malaysia in 2016. Her research interests include nature-inspired algorithms, optimization problems, data mining and business intelligence.

**Abu Bakar Md. Sultan** is a professor and dean in Faculty of Computer Science and Information Technology at University Putra Malaysia. He completed his Ph.D in intelligent system at University Putra Malaysia in 2006. His research interests include metaheuristics, evolutionary and computing.

**Md. Nasir Sulaiman** is an associate professor in Faculty of Computer Science and Information Technology at Universiti Putra Malaysia. He completed his Ph.D in computer science at Loughborough University, U.K., in 1994. His research interests include intelligent computing and data mining.

**Aida Mustapha** is a senior lecturer in Faculty of Computer Science and Information Technology at University Tun Hussein Onn Malaysia. She is also the deputy dean of research and development at the same university. She completed her Ph.D in computational linguistics at University Putra Malaysia in 2008. Her research interests include stylometry, soft computing, and spoken dialogue systems.

**K.Y. Leong** is a Ph.D candidate. Currently, he is pursuing his Ph.D at Monash University Malaysia. His research project mainly focuses on cognitive robotics and assistive technology in mobility for the visually impaired people.