An Overview on the Utilization of Self-Adaptive Differential Evolution

Sarah Hazwani Adnan
Centre of Global Sustainability Studies, Universiti Sains Malaysia, 11800, Penang, Malaysia

Shir Li Wang
Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, 35900, Perak, Malaysia

Haidi Ibrahim
School of Electrical and Electronic Engineering, Universiti Sains Malaysia, 11800, Penang, Malaysia

Theam Foo Ng
Centre of Global Sustainability Studies, Universiti Sains Malaysia, 11800, Penang, Malaysia, tfng@usm.my

Abstract

Different Evolution or DE is possibly the most current powerful optimization algorithm and has been used in multiple diverse area such as neural networks, logistic, scheduling, modeling and others. DE has been widely known for its simplicity, ease of implementation and reliability in getting an optimal solution. However, different problem requires different parameter settings and thus, it became quite a challenge for DE to select appropriate parameter setting in tackling complex computational optimization problem. Self-Adaptive Differential Evolution or SADE is an improved version of DE, intended to simplify the process of choosing the most suitable parameter to be used for solving problems. Carrying the strength of DE, which is as an efficient and robust optimizer technique, this ability of SADE has attracted a lot of research effort in the past years by simplifying the process of fine tuning the most suitable parameter to be used. With the introduction of SADE in optimization areas, where the choice of learning strategy and parameter setting do not require predefining, parameter tuning has become less confusing. SADE is gradually becoming one of the popular research topics, mainly in computer science, engineering and mathematics area. The algorithm has also been applied in numerous disciplines such as electromagnetic, power system, computer performance, fermentation, polyester process and more. SADE has also proven to achieve better performance compared to a conventional DE algorithm. This paper aims at providing an overview on the utilization of SADE from year 2015 to 2017. By collecting and analyzing related articles that have implemented SADE in solving problems, a significant trend in the usage of SADE is provided.

Keywords: *Algorithm, optimization, overview, self-adaptive differential evolution, SADE*

1. Introduction

Differential Evolution (DE), introduced by Storn and Price in 1995, has been known as a reliable, robust, simple and straightforward to implement, fast optimization technique [1]-[3], powerful search algorithm and effective evolutionary optimization algorithm (*EA*) [4], [5]. The algorithm has three control parameters, which are mutation control parameter (*F*), crossover control parameter (*CR*) and population size (*NP*), where the values of these parameters greatly determine the quality of the solution obtained and the efficiency of the search [6], [7]. Choosing suitable value for each of the parameters is usually problem-dependent task. No strict rule imposed in searching for the most suitable value ranges of F and *CR* [8] where it is often arbitrarily set within some predefined ranges [9], [10]. In order to make sure the algorithm able to choose the most suitable parameters for each problem, several researchers had investigated on the self-adaptive algorithm.

As DE is an improved version of Genetic Algorithm (GA) [11], self-adaptive algorithm is an extension of DE. There are several attempts to control the parameter using self-adaptive algorithm, where mostly focus on simplifying the DE process of finding the perfect parameter to be used. Self-adaptive Differential Evolution or SADE introduce the idea of adaptively change values of *F* and *CR* instead of taking fixed values [12]. While Self-adaptive Differential Evolution or SDE use normal distribution of different means and standard deviations [13]. FADE or Fuzzy Adaptive Differential Evolution introduce the use of fuzzy logic in order to dynamically control *F* and *CR* values [14].

In this paper, we are using "SADE" to address multiple referring name or abbreviation to self-adaptive differential evolution. The purpose of this paper is to provide an overview on the utilization of SADE. We restrict our overview of articles published from year 2005 until 2017, where the articles either are using SADE as a benchmark for their developed algorithm, manipulating SADE performance or implementing SADE on specific problem.

2. Scope of Overview

The literature search is done by evaluating recent articles, published between 2005 and 2017. As the SADE algorithm is developed in 2005, we start to find any related articles that implement the algorithm starting from 2005. In this literature overview, only relevant articles published in Scopus are considered except books, thesis and dissertations.

A certain search strategy was applied in order to choose the most related articles for this literature. Only articles containing "self-adaptive differential evolution" or "adaptive differential evolution" in the article title were selected. However, for an article that has only one or two of the keyword searches, it has the chance to be included as long as the algorithm follows the same rules as SADE. The algorithm needs to be able to self-adapt, and no need for pre-determined mutation parameter and crossover rate. This search strategy resulted in 348 articles. All of the found articles are categorized into three categories of utilization to enable us analyze the trend of SADE easily in current research works.

3. Result of Overview

It is found that the utilization of SADE can be categorized into three categories, which are Modification and Implementation – applying the SADE algorithm in solving any problem, Performance Comparison - comparing the performance of author's developed algorithm with SADE and Performance Enhancement - algorithm enhancement in order to boost the algorithm performance itself.

Modification and Implementation category grouped researches that are using the concept of self-adaptive DE into real life problems. Performance Comparison category grouped researches that are using another algorithm or related adaptive algorithm to solve intended problems and compare the result to self-adaptive algorithm. Whereas Performance Enhancement category grouped researches that alter self-adaptive algorithm without trying to solve real-life problems and tested it with benchmark data. There are also multiple researches that tweak self-adaptive algorithm and later on implements it on real-life problems. For that type of research, it will be considered into modification and implementation type of problems. There are also researches that can be categorized into more than one category, such as [8] and [15]. However, we have categorized all of the researches into the most suitable category in this overview.

SADE has been introduced in 2005 by [13] under the name SDE and has successfully outperformed the other DE functions in all benchmark function experiments. Since then, many authors have tried to adapt SADE into numerous distinct problems, with the majority falls under the optimization area. Believing that the algorithm is strong enough to be used as it is, that is probably why most of the articles published are using SADE algorithm to help in solving author's intended problem. Figure 1 shows the trend of SADE implemented articles from year 2005 until 2017. The number of articles under Modification and Implementation category begins with gently rising in number to steeply increase starting from 2012. The category has 240 articles which are for about four times more than the articles listed under Performance Enhancement category which is only 65 articles. The number of articles for Performance Enhancement category shows a noticeable change for about a decade where it keeps on raising gradually. Most of the articles for this category had able to produce an acceptable enhancement result. The Performance Comparison category has lesser articles compared to the other two categories, but the number of articles still rise lightly towards the end of 2017, which are 43 articles. Overall, we can see a clear upward trend in all three categories for the number of articles published with implementing SADE algorithm.

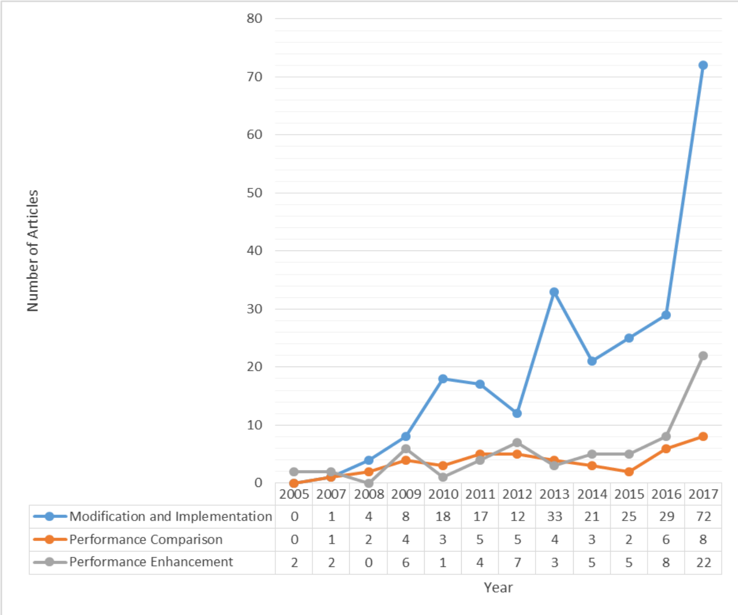


Figure 1. Number of reviewed articles categorized into three different types of categories which implementing SADE since year 2005 until 2017.

3.1 Modification and Implementation

Most of the articles published related to SADE fall under the implementation of the algorithm. According to Figure 1, we can see that the trend of using self-adapted algorithm is more on its implementation. Most of the researches done had produced a positive impact on the performance of the algorithm towards the problem. As the descendant of DE algorithm, SADE is also recognized as a robust optimization algorithm, able to outperform classical DE and flexible as it already being used in multiple distinct disciplines.

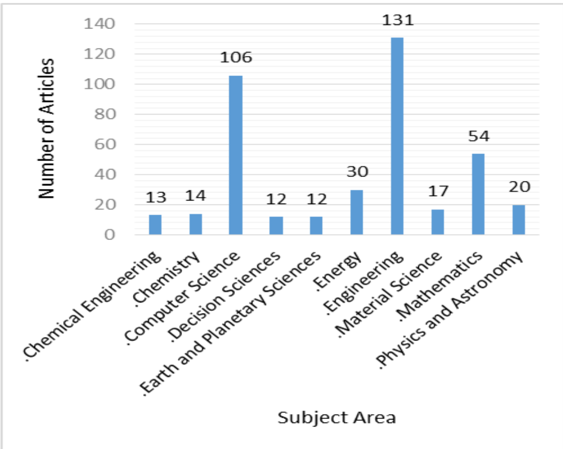


Figure 2. Top 10 areas for modification and implementation category.

By referring to Figure 2, SADE is mostly employed in engineering, computer science, and mathematical modeling subjects. Where there are few that had been implemented in mathematical subject and social science subject, the objective is the same; which is to obtain the most optimum solution for the problem. SDE by Al-anzi and Allahverdi [16] is first implemented in the flowshop scheduling problem in order to minimize maximum lateness which can also be used for other related problems. In 2008, the algorithm is implemented in power economic dispatch problem [17], flowshop scheduling problem [18], tuning of a chess program [19], microwave absorber design [20] and security constrained optimal power flow [21].

The ways control parameter can be made self-adaptive have regained interest in many other problems. In 2009, SADE has successfully provided a better solution for energy problem [22], [23], nuclear division

[24] and in transportation design problem [24]-[26]. Most of the given solutions showed that self-adaptive differential evolution outperforming in benchmark problems [27]-[30]. There were also cases where SADE did not perform well or increase the number of function evaluations [31]. Despite that, SADE had been employed in different types of engineering problems [32]-[37], biology related problems [38]-[42] and power system related problem [43]-[47].

3.2 Performance Comparison

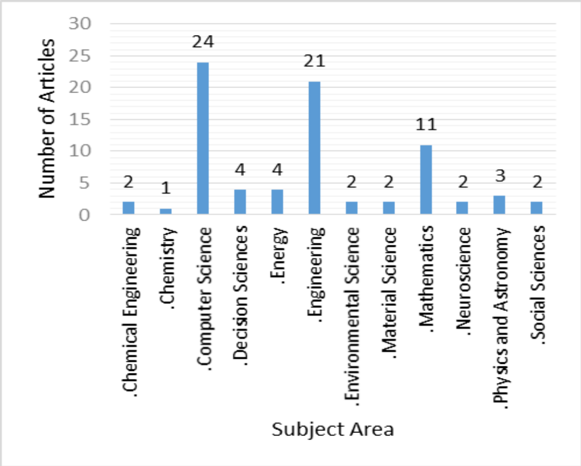


Figure 3. Top 10 areas for performance comparison category.

SADE has been used as a benchmark algorithm for multiple problems. Similar with Modification and Implementation category, SADE has also been highly adapted in Computer Science, Engineering and Mathematics related area as shown in Figure 3. Considering the algorithm as a stable, robust and an efficient optimization algorithm, it has been chosen as one of the established algorithm to be a yardstick for optimization problems. However, not many research that used SADE as comparison towards their proposed algorithm. Most of researches found in this category able to produce new or improved algorithm where it is significantly better or least acceptable performance compared to using SADE algorithm. However, as a successor of Differential Evolution, self-adaptive concept has gained more popularity and is commonly used to solve problem compared to other types of algorithms, such as swarm algorithm and genetic algorithm [48].

Implemented in yeast fermentation process, proposed technique had shown to be more successful by integrating self-adaptive mechanism. There were also cases where SADE did not produce a good result [49], [50] in some test [50] but that does not mean that SADE is a failure. More than 30 articles found are tested in gene sorting problem [51] and benchmark functions [52]-[56]. Algorithm integrated with SADE produced better results than previous used algorithms for the tested problems [48].

3.3 Performance Enhancement

As shown in Figure 4, Computer Science area has lots of articles which enhance the SADE algorithm where the majority is applied in solving optimization problem. Being developed to counter the drawback of DE, the idea of auto-tuning two critical parameters of DE [13], [14] has greatly evolved and implemented in multiple diverse areas. SADE algorithm is developed in order to simplify DE process of finding the appropriate parameters to be used [2]. Liu and Lampinen [14] had used fuzzy knowledge based system in order to auto-adapt the search parameter for mutation and crossover operation. However, fuzzy rules are dependent to expert's knowledge as it is using human knowledge and previous experience in order to construct fuzzy rules and membership functions [1]. Whereas Omran *et al*. [13] auto-adapt the search parameter by using normal distributions of different means and standard deviation. This approach is quite similar with SPDE in Abbas research [13], [28]. However, SPDE used a normal distribution with mean zero and standard deviation one (N(0,1)) for F [13]. These two articles can be considered as the pioneer of self-adaptive strategy. Nobakhti and Wang [30] introduce co-evolutionary process into the process of controlling F. With reference to the balance of exploitation versus exploration, three parameters from co-evolutionary process will govern F value such that it can help in preventing stagnation or stopping the mutation [30].

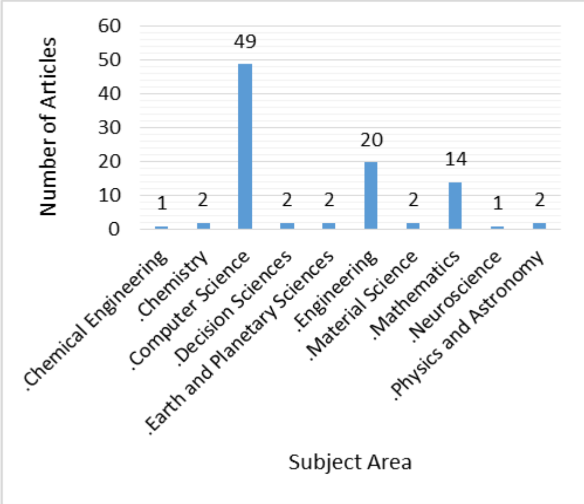


Figure 4. Top 10 areas for performance enhancement category.

Other related work had been done by applying the immune system concept in determining parameter of DE [28]. As a highly evolved biological system with learning, memory and pattern recognition capability, immune system proposed, named ISDE, had used previous information about search to tune and seek the optimal parameters *F* and *CR*. By manipulating previous information gather, ISDE can rightfully be regarded as a reliable optimization algorithm. In JADE, two new parameters are introduced. The c parameter controls the rate of adaptation, whereas p parameter determines the greediness of mutation strategy [57]. Brest *et al*. have proposed boundary constraints value in the jDE and jDE-2 algorithms. jDE-2 has shown to be better than the former because it uses two DE strategies compared to jDE which is based on only one DE strategy [58].

Adaptation of self-adaptive strategy has also been confirmed in MOSADE experiment that it is able to greatly improve the robustness of the algorithm [59]. In p-ADE, dynamic self-adaptation strategy is applied into DE/rand-to-best/pbest mutation strategy where it is beneficial in controlling the extent of variation for each individual [60]. In SADE-MMTS, the author enhance SADE algorithm by integrating JADE mutation strategy (DE/current-to-pbest with small-size archive named arcs) and modified multiple trajectory search (MMTS) [61]. The effectiveness of self-adjusted parameter based on previous knowledge in adaptive parameter concept of SADE had also captivate another author into employing this concept with extreme learning machine (ELM) as SaE-ELM [62] and with α-constrained-domination principle as SADE-αCD [63]. Another author had come up with the idea of introducing two self-adaptive DE algorithms which both incorporate a heuristic mixing of operators (DE/HMO) [64]. Both variants split the population into sub-population and local search procedures are used to speed up the convergence of the solution. Self-adaptive parameter is an interesting and efficient technique introduced. That is why most of the articles found are manipulating this algorithm by integrating it with another algorithm [15], [65]-[70].

4. Conclusion

There are 348 articles found related to implementation of Self-Adaptive Differential Evolution in solving different problems. All of these articles can be differentiated into 3 main categories which are modification and implementation of SADE techniques (Modification and Implementation), SADE as a benchmark for another algorithm (Performance Comparison) and SADE performance’s enhancement (Performance Enhancement). Sometimes, the article can fall into more than one category listed. However, from the best of our knowledge, by looking through all of the articles, the view is still the same in which manipulation of SADE can be categorized in these three categories.

In handling optimization problem, there are certain critical parameters that can give a major effect towards the problem. As an enhanced version of DE, SADE had helped in managing critical parameters lies in DE which are *F* and *CR*.

SADE has proven to be capable of producing a reliable and better results, with ease of implementation in multiple diverse areas, e.g., applied energy, mathematical application, wireless propagation related problems, chemistry, robotics, operational research, biology related problems and more. Whereas, computer science, engineering and mathematics area the most popular area for implementing the SADE algorithm as shown in Table 1. Therefore, many authors had chosen SADE as one of the benchmarks in testing new algorithm.

Table 1. Top 10 Journal that Are Publishing Articles Implementing SADE

No	Journal	No of articles
1	Soft Computing	16
2	Applied Soft Computing Journal	11
3	Computers and Operations Research	7
4	Lecture Notes in Computer Science	7
5	Expert Systems with Applications	6
6	Kongzhi Yu Juece/Control and Decision	6
7	International Journal of Electrical Power and Energy Systems	5
8	Neurocomputing	5
9	Swarm and Evolutionary Computation	5
10	Applied Intelligence	4

Most of the manipulation made on this algorithm is solely on adapting the algorithm into another algorithm. We can say that many authors have confidence in the performance of the algorithm and they are able to accept the concept of self-adaptive parameter which is already robust and independent, and can be used simply as it is. Only a small number of articles intended to change the SADE algorithm [64], [66] as shown in the Performance Enhancement category.

Future research could explore another way to speed up F and CR processes, or manipulating the algorithm by introducing another parameter to be self-adapted. Survey on simulation implementing SADE can also be done as analyzing the value of self-adapt parameters for multiple problems is still essential for the growth and robustness of the algorithm itself. Another thorough overview on the application of SADE can be made as there are many underlying issues that we have not covered.

Acknowledgment

The authors would like to acknowledge and thank the Universiti Sains Malaysia and the Ministry of Higher Education, Malaysia for supporting this research through the Fundamental Research Grant Scheme with account number 203/PCGSS/6711501.

References

[1] A. Salman, A. P. Engelbrecht, and M. G. H. Omran, "Empirical analysis of self-adaptive differential evolution," *European Journal of Operational Research*, vol. 183, no. 2, pp. 785-804, 2007.

[2] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398-417, 2009.

[3] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4-31, 2011.

[4] H. Fu, D. Ouyang, and J. Xu, "A self-adaptive differential evolution algorithm for binary CSPs," *Computers & Mathematics with Applications*, vol. 62, no. 7, pp. 2712-2718, 2011.

[5] X. Lu, K. Tang, B. Sendhoff, and X. Yao, "A new self-adaptation scheme for differential evolution," *Neurocomputing*, vol. 146, pp. 2-16, 2014.

[6] T. Krink and R. K. Ursem, "Parameter control using the agent based patchwork model," in *Proc. 2000 Congress on Evolutionary Computation*, 2002, pp. 77-83.

[7] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646-657, 2006.

[8] C. Chellaswamy and R. Ramesh, "Parameter extraction of solar cell models based on adaptive differential evolution algorithm," *Renewable Energy*, vol. 97, pp. 823-837, 2016.

[9] M. H. Maruo, H. S. Lopes, and M. R. Delgado, "Self-adapting evolutionary parameters: Encoding aspects for combinatorial optimization problems," in *Proc. EvoCOP 2005*, 2005, pp. 154-165.

[10] S. T. Pan, "Evolutionary computation on programmable robust IIR filter pole-placement design," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 4, pp. 1469-1479, 2011.
[11] A. Vasan and K. S. Raju, "Optimal reservoir operation using differential evolution," *International Conference on Hydraulic Engineering: Research and Practice (ICON-HERP-2004),* pp. 1-12, 2004.
[12] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1785-1791.
[13] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Self-adaptive differential evolution," in *Proc. 2005 International Conference on Computational Intelligence and Security*, 2005, vol. 1, pp. 192-199.
[14] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448-462, 2004.
[15] Z. Yan, L. Zhaobin, and B. Zhang, "A novel differential evolution algorithm for constrained optimization," in *Proc. IEEE International Conference on Computational Science & Engineering*, 2017, pp. 342-348.
[16] F. S. Al-Anzi and A. Allahverdi, "A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times," *European Journal of Operational Research*, vol. 182, no. 1, pp. 80-94, 2007.
[17] L. S. Coelho and V. C. Mariani, "Self-adaptive differential evolution using chaotic local search for solving power economic dispatch with nonsmooth fuel cost function," *Advances in Differential Evolution*, pp. 275-286, 2008.
[18] A. Allahverdi and F. S. Al-Anzi, "The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time," *The International Journal of Advanced Manufacturing Technology*, vol. 37, no. 1-2, pp. 166-177, 2007.
[19] B. Boskovic, S. Greiner, J. Brest, A. Zamuda, and V. Zumer, "An adaptive differential evolution algorithm with opposition-based mechanisms, applied to the tuning of a chess program," *Advances in Differential Evolution*, pp. 287-298, 2008.
[20] S. K. Goudos, "Design of microwave broadband absorbers using a self-adaptive differential evolution algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 19, no. 3, pp. 364-372, 2009.
[21] C. Thitithamrongchai and B. Eua-Arporn, "Security-constrained optimal power flow: A parallel self-adaptive differential evolution approach," *Electric Power Components and Systems*, vol. 36, no. 3, pp. 280-298, 2008.
[22] R. Balamurugan and S. Subramanian, "Emission-constrained dynamic economic dispatch using opposition-based self-adaptive differential evolution algorithm," *International Energy Journal 10*, pp. 267-276, 2009.
[23] Z. Yang, Z. I. Rauen, and C. Liu, "Automatic tuning on many-core platform for energy efficiency via support vector machine enhanced differential evolution," *Scalable Computing: Practice and Experience*, vol. 18, no. 2, pp. 117-131, 2017.
[24] G. K. Pierens, M. Mobli, and V. Vegh, "Effective protocol for database similarity searching of heteronuclear single quantum coherence spectra," *Analytical Chemistry*, vol. 81, no. 22, pp. 9329-9335, 2009.
[25] A. Koh, "An adaptive differential evolution algorithm applied to highway network capacity optimization," *Applications of Soft Computing*, vol. 52, pp. 211-220, 2009.
[26] D. Adhikari, E. Kim, and H. Reza, "A fuzzy adaptive differential evolution for multi-objective 3D UAV path optimization," in *Proc. 2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 2258-2265.
[27] D. Jing, C. Jian, and S. Min, "Cooperative task assignment for heterogeneous multi-UAVs based on differential evolution algorithm," *Intelligent Computing and Intelligent Systems*, pp. 163-167, 2009.
[28] C. Hu and X. Yan, "An immune self-adaptive differential evolution algorithm with application to estimate kinetic parameters for homogeneous mercury oxidation," *Chinese Journal of Chemical Engineering*, vol. 17, no. 2, pp. 232-240, 2009.
[29] J. Tvrdík, "Adaptation in differential evolution: A numerical comparison," *Applied Soft Computing*, vol. 9, no. 3, pp. 1149-1155, 2009.
[30] A. Nobakhti and H. Wang, "A simple self-adaptive differential evolution algorithm with application on the ALSTOM gasifier," *Applied Soft Computing*, vol. 8, no. 1, pp. 350-370, 2008.
[31] A. Chowdhury, A. Ghosh, R. Giri, and S. Das, "Optimization of antenna configuration with a fitness-adaptive differential evolution algorithm," *Progress in Electromagnetics Research B*, vol. 26, pp. 291-319, 2010.

[32] R. D. Al-Dabbagh, A. Kinsheel, S. Mekhilef, M. S. Baba, and S. Shamshirband, "System identification and control of robot manipulator based on fuzzy adaptive differential evolution algorithm," *Advances in Engineering Software*, vol. 78, pp. 60-66, 2014.
[33] Q. Lei, M. Wu, and J. She, "Online optimization of fuzzy controller for coke-oven combustion process based on dynamic just-in-time learning," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1535-1540, 2015.
[34] K. R. Vadivelu and G. V. Marutheeswar, "Optimal reactive power planning using improved differential evolution incorporating facts," *Journal of Electrical Engineering*, pp. 1-8, 2015.
[35] Z. Li, W. Du, L. Zhao, and F. Qian, "Synthesis and optimization of utility system using parameter adaptive differential evolution algorithm," *Chinese Journal of Chemical Engineering*, vol. 23, no. 8, pp. 1350-1356, 2015.
[36] A. W. A and R. Rengaraj, "Economical operation of thermal generator involving transmission loss using noval capra optimization algorithm," *Journal of Electrical Engineering*, pp. 1-10, 2016.
[37] M. Salehpour, A. Jamali, A. Bagheri, and N. Nariman-zadeh, "A new adaptive differential evolution optimization algorithm based on fuzzy inference system," *Engineering Science and Technology, An International Journal*, vol. 20, no. 2, pp. 587-597, 2017.
[38] Y. Tang, H. Gao, W. Du, J. Lu, A. V. Vasilakos, and J. Kurths, "Robust multiobjective controllability of complex neuronal networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 4, pp. 778-791, 2016.
[39] M. Kabir, N. Noman, and H. Iba, "Reverse engineering gene regulatory network from microarray data using linear time-variant model," *BMC Bioinformatics*, vol. 11, no. 1, pp. 1-15, 2010.
[40] J. Zhang, V. Avasarala, and R. Subbu, "Evolutionary optimization of transition probability matrices for credit decision-making," *European Journal of Operational Research*, vol. 200, no. 2, pp. 557-567, 2010.
[41] S. Sudha, S. Baskar, S. M. J. Amali, and S. Krishnaswamy, "Protein structure prediction using diversity controlled self-adaptive differential evolution with local search," *Soft Computing*, vol. 19, no. 6, pp. 1635-1646, 2014.
[42] W. Gómez, W. C. A. Pereira, and A. F. C. Infantosi, "Evolutionary pulse-coupled neural network for segmenting breast lesions on ultrasonography," *Neurocomputing*, vol. 175, pp. 877-887, 2016.
[43] X. Zhang, W. Chen, C. Dai, and W. Cai, "Dynamic multi-group self-adaptive differential evolution algorithm for reactive power optimization," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 5, pp. 351-357, 2010.
[44] H. Beirami, A. Z. Shabestari, and M. M. Zerafat, "Optimal PID plus fuzzy controller design for a PEM fuel cell air feed system using the self-adaptive differential evolution algorithm," *International Journal of Hydrogen Energy*, vol. 40, no. 30, pp. 9422-9434, 2015.
[45] Y. Chen, F. Luo, Y. Xu, and J. Qiu, "Self-adaptive differential approach for transient stability constrained optimal power flow," *IET Generation, Transmission & Distribution*, vol. 10, no. 15, pp. 3717-3726, 2016.
[46] F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, "Evolutionary algorithms for power generation planning with uncertain renewable energy," *Energy*, vol. 112, pp. 408-419, 2016.
[47] X. Zhang, "Population-adaptive differential evolution-based power allocation algorithm for cognitive radio networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1-8, 2016.
[48] S. C. Wang, "Differential evolution optimization with time-frame strategy adaptation," *Soft Computing*, vol. 21, no. 11, pp. 2991-3012, 2016.
[49] S. Roy, S. M. Islam, S. Ghosh, and S. Das, "An adaptive differential evolution algorithm for autonomous deployment and localization of sensor nodes," *Progress in Electromagnetics Research B*, vol. 29, pp. 289-309, 2011.
[50] J. Teo, "Analyzing the scalability performance of crossover-first and self-adaptive differential evolution algorithms for complex numerical optimization," *International Journal on Advances Dcience Engineering Information Technology*, vol. 7, no. 5, pp. 1847-1852, 2017.
[51] R. Tassing, L. Guo, J. Liu, H. Lin, and G. Zhu, "Gene sorting in differential evolution with cross-generation mutation," *Science China Information Sciences*, vol. 54, no. 2, pp. 268-278, 2011.
[52] W. P. Lee and C. Y. Chiang, "A self-adaptive differential evolution algorithm with dimension perturb strategy," *Journal of Computers*, vol. 6, no. 3, pp. 524-531, 2011.
[53] J. Brest and M. S. Maučec, "Self-adaptive differential evolution algorithm using population size reduction and three strategies," *Soft Computing*, vol. 15, no. 11, pp. 2157-2174, 2010.

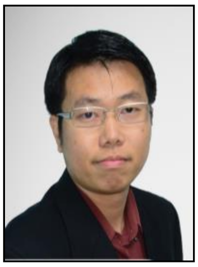
[54] C. C. Chiu and C. H. Sun, "Computational approach based on a differential evolution with self-adaptive concept for microwave imaging of two-dimensional inverse scattering problem," *Electromagnetics*, vol. 32, no. 8, pp. 451-464, 2012.
[55] M. Bashiri, H. Vatankhah, and S. S. Ghidary, "Hybrid adaptive differential evolution for mobile robot localization," *Intelligent Service Robotics*, vol. 5, no. 2, pp. 99-107, 2012.
[56] S. M. Venske, R. A. Gonçalves, and M. R. Delgado, "ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm," *Neurocomputing*, vol. 127, pp. 65-77, 2014.
[57] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945-958, 2009.
[58] J. Brest, "Constrained real-parameter optimization with ε-self-adaptive differential evolution," *Studies in Computational Intelligence*, pp. 73-93, 2009.
[59] Y. N. Wang, L. H. Wu, and X. F. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," *Soft Computing*, vol. 14, no. 3, pp. 193-209, 2009.
[60] X. J. Bi and J. Xiao, "Classification-based self-adaptive differential evolution with fast and reliable convergence performance," *Soft Computing*, vol. 15, no. 8, pp. 1581-1599, 2011.
[61] S. Z. Zhao, P. N. Suganthan, and S. Das, "Self-adaptive differential evolution with multi-trajectory search for large-scale optimization," *Soft Computing*, vol. 15, no. 11, pp. 2175-2185, 2010.
[62] J. Cao, Z. Lin, and G. B. Huang, "Self-adaptive evolutionary extreme learning machine," *Neural Processing Letters*, vol. 36, no. 3, pp. 285-305, 2012.
[63] F. Qian, B. Xu, R. Qi, and H. Tianfield, "Self-adaptive differential evolution algorithm with α-constrained-domination principle for constrained multi-objective optimization," *Soft Computing*, vol. 16, no. 8, pp. 1353-1372, 2012.
[64] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Self-adaptive differential evolution incorporating a heuristic mixing of operators," *Computational Optimization and Applications*, vol. 54, no. 3, pp. 771-790, 2012.
[65] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with discrete mutation control parameters," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1551-1572, 2015.
[66] C. Brown, Y. Jin, M. Leach, and M. Hodgson, "μJADE: Adaptive differential evolution with a small population," *Soft Computing*, vol. 20, no. 10, pp. 4111-4120, 2015.
[67] H. Guo, Y. Li, J. Li, H. Sun, D. Wang, and X. Chen, "Differential evolution improved with self-adaptive control parameters based on simulated annealing," *Swarm and Evolutionary Computation*, vol. 19, pp. 52-67, 2014.
[68] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 219-232, 2016.
[69] H. Peraza-Vázquez, A. M. Torres-Huerta, and A. Flores-Vela, "Self-adaptive differential evolution hyper-heuristic with applications in process design," *Computación y Sistemas*, vol. 20, no. 2, pp. 173-193, 2016.
[70] W. Gong, Z. Cai, and Y. Wang, "Repairing the crossover rate in adaptive differential evolution," *Applied Soft Computing*, vol. 15, pp. 149-168, 2014.

Sarah Hazwani Adnan received a bachelor degree in computer science (hons.) from Universiti Teknologi Mara (UiTM), Seri Iskandar Campus, Perak, Malaysia in 2014 and are currently pursuing her master degree in science (sustainability) focusing on computer modelling from Universiti Sains Malaysia (USM), Pulau Pinang, Malaysia.

Shir Li Wang obtained her PhD degree at University of New South Wales, Canberra Campus, Australia in 2012. She received the BSc degree in 2002 and MSc degree in 2007, both from the Universiti Sains Malaysia. She is currently is a senior lecturer at Faculty of Art, Computing and Industry Creative, Sultan Idris Education University. Her research interests focus on artificial intelligence specifically evolutionary computation, neural network and clustering, as well as their usage in problem solving such as optimization, prediction and modelling. Her interest and passion in the

artificial intelligence motivates her to further explore the potential of artificial intelligence in education.

Haidi Ibrahim received his B.Eng degree in electronic and electrical engineering from Universiti Sains Malaysia, Malaysia. He is then received his Ph.D degree in image processing from Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, United Kingdom. His research interests are in digital image processing and analysis, including image enhancement, noise reduction, image segmentation, 3D visualization, and virtual reality.



Theam Foo Ng obtained his B.Sc (hons) mathematics in 2002, and M.Sc (statistics) in 2006 from Universiti Sains Malaysia (USM), Penang, Malaysia. He received his Ph.D degree from the University of New South Wales, Australia in 2012. He is now with the Centre for Global Sustainability Studies (CGSS), USM, Penang, Malaysia, where he is currently as a senior lecturer. His research interest is in the area of education for sustainable development, disaster risk management, machine learning, pattern recognition, computational intelligence, artificial intelligence, image processing and mathematics.