

Self-adaptive Ensemble Based Differential Evolution

Shir Li Wang, Theam Foo Ng, and Farid Morsidi

Abstract—Differential evolution (DE) is among the more prominent branch of evolutionary algorithm (EA) innovated for multiple optimization properties. It has been improvised in various practical solutions, whether it is for benchmark testing or real world situations. As compared with other stochastic optimization algorithms such as nature inspired algorithms and evolutionary ones, DE possesses savvy traits in terms of exploration and exploitation within its own domain. With its motives of locating optimal points and minimized solution steps for objective functions, DE relied heavily on the necessity to specify parameter settings that is catered for achieving appropriate convergence values. The exhibited parameter value is seen directly correlated with the quality of the solutions for the underlying optimization problem. However, selection of appropriate parameter values occasionally necessitate for a priori experience and problem dependent on user. In most cases, users emphasize more on solving the optimization problem rather than solving the algorithm itself. Besides that, research work related to parameter study in DE lacks of proper and clear guidance to users. Therefore, there is a need to develop a DE which can adaptively determine the appropriate parameters to solve different optimization problems with minimum guidance from users. In this research, we take the opportunity to develop a DE model which combines self-adaptive and ensemble mechanisms to dynamically change the control parameters as well as mutation strategy during evolution with minimum intervention from users. The experimental results have shown that the proposed model is able to perform adequately well in twenty different benchmark problems without depending on user to determine the parameters explicitly.

Index Terms—Differential evolution, parameters, mutation strategy, self-adaptive.

I. INTRODUCTION

A consistent pattern of trend rising in consecutive years since the introduction of DE into the field of EAs by Storn and Price [1] towards rectifying problems associating with optimization problems, where among the main factors that contribute to this could be seen in the robust, simplicity and reliability that DE constitutes to deal with non-differentiable and multi-modal optimization problems. Just like any EA, the effectiveness of DE relies on the setting on two main components, i.e., mutation strategies and control parameters. Mutation strategies refers to ways that population vectors

being selected to go through differential mutation process. Control parameters refers to population size, scale factor and crossover rate. In this regard, the setting of mutation strategies and control parameter influences the generation of potential population vectors which are the potential solutions of an optimization problem. Therefore, users who like to use DE to solve an optimization problem need to first find the good settings for the control parameters as well as mutation strategy.

To use DE appropriately, users need to equip themselves with some knowledge and experiences because selection of relevant parameters in DE is very task dependent [2], [3]. Even guidance is provided in selecting control parameters of DE, but the guidance could either be lacking in justification or causing conflicting results [2]. Besides that, most of the researches focus specifically on the setting of crossover rate and scale factor but only a few are on population size. The research on setting mutation strategy is even less. Therefore, the common approach adopted by DE users is the trial-and-error method [2], [3]. The approach is time consuming but there is no guarantee that the optimal solutions are obtained. In reality, most of the DE users are interested to solve the underlying optimization problem only rather than the algorithm itself. Therefore, the main objective of our research it to find a solution that enable DE to operate by itself with minimum intervention from users in parameter setting but still perform adequately. This means that the proposed DE can adaptively determine the appropriate population size, crossover rate scale factor as well as mutation strategy to solve different optimization problems.

Another potential advantage of our proposed DE is that the model could still produce optimal solutions as close as possible to the global optimal solutions even if they have been shifted or changed over time. When there is a change in the underlying optimization problem, the previous set of parameters may not be appropriate anymore to find the shifted global optimal points. In our proposed DE model, we use both self-adaptive and ensemble mechanisms to find the appropriate set of control parameters and mutation strategy. The implementation of the mechanisms are described in Section III. With the proposed mechanisms, different values of population size, crossover rate scale factor as well as mutation strategy are used in different stages of evolution. Owing to this, we are interested to know that whether the proposed DE model can still perform as adequate of some of the existing models such as differential evolution in relative coding (DE-Rel) and an ensemble of mutation strategies and control parameters with the differential evolution (ESPDE).

The rest of this paper is organized as follows. Section II describes related work to DE. Section III presents the methodology of our proposed DE model and Section IV describes the experimental setup. Discussion of the experimental results through the analysis methodology

Manuscript received January 5, 2018; revised March 17, 2018.

Both Shir Li Wang and Farid bin Morsidi are with the Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, Tanjung Malim, 35900, Perak Malaysia (e-mail: shirli_wang@fskik.upsi.edu.my, M20142001428@siswa.upsi.edu.my).

Theam Foo Ng is with Centre for Global Sustainability Studies, Universiti Sains Malaysia, 11800, Penang, Malaysia (e-mail: tfng@usm.my).

follows in Section V. Finally, Section VI presents conclusions and future work.

II. LITERATURE REVIEW

Computational intelligence-based techniques is regarded as practical and powerful solution schemes to obtain the global or quasi-global optimum solution to engineering optimization problems [4]. There is a rise in need for algorithms that perform well on designating multiple problems simultaneously [5]. EA is adept in identifying multiple solution within a population in a single simulation run and possesses a clear advantage over the classical optimization technique [6]. It can be used to resolve optimization problems which can be either single objective, multi-objectives and constrained optimization problems.

DE has an edge over relative EAs in terms of floating-point encoding aside with lead combinations of evolutionary operators, mutation step size adaptation, and elitist selection [7]. The success of DE is contributed by its two-phase operation, i.e., exploration and exploitation. Exploration is about exploring the search space for good solutions, while exploitation is about refining the solutions. The two-phase operation highly depends on the parameter setting of crossover rate (CR) and scale factor (F).

Parameter setting is a debatable subject that is crucial in propagating better values for discovering proper solution steps. For the control parameters CR and F , there is no so called distinctive rules to set their values but the setting affects the functionality of DE [8]. For example, a large value of F increases the exploration ability but decreases the exploitation ability and vice versa. In contrast, an increase in CR deteriorates the quality of optimal solutions while a decrease causes the algorithm to be stagnant. The importance F and CR in affecting the convergence velocity and robustness of the search process is tested by preserving a fixed population size and scale factor to prevent premature convergence and stagnation [9].

A variant DE algorithm modified slightly at mutation, crossover, and selection is applied to analyze the relationship between control parameters for DE algorithms and the evolution of population variance from theoretical and empirical viewpoint [10]. The poor behavior of DE for multimodal problems is also addressed by observation on the crossover types on the evolution steps behavior [10].

The work in [11] had summarized self-adaptive ensemble in assessing multiple optimization and redesign when it is assigned to real world problems or other evolutionary algorithms. It is interesting to note that population size (NP) and constraint handling methods plays a pivotal role in determining superiority of feasible solutions and the speed of convergence rate in which it occurs across dynamic dimensionality and ranges. However, an increase in NP takes more objective evaluations to generate the same effect [12]. Adaptive population tuning scheme for DE had been proposed to dynamically adjust the population size based on the solution-searching status, the desired population distribution for children generation strategies and parameter adaptation mechanisms [13]. However, the adaptive population tuning scheme involves more user-defined variables.

The nature of optimization problems such as noise and high dimensionality also affects parameter setting [8]. Higher dimensionality in an optimization problem requires higher values of NP . Dealing with noisy optimization problems, a constant F for the whole evolution is not sufficient to search for good solutions [14]. Using low values of CR when the objective function is multi-modal and non-separable can cause the search to take longer than a simple random search [15]. However in reality, users hardly know in advance the nature of the problems that they deal with.

Exploration of parameter adaptation schemes has been emphasized in overcoming deficiencies in mutation strategies, such as premature convergence, stagnation, and computational resource wastage [16]. It is shown in [2] that dynamic setting of control parameters as well as mutation strategy are required in different stages of evolution to ensure adequate performance.

One of the procedures introduced in tackling dynamic parameter adaptation for optimizing solution steps in DE is via inducing self-adaptive strategy on structural components. This feature is generated in terms of changeover for structure, flow, or genetic component. Among the selected few approaches that execute self-adaptive methodology are summarized by [5], [17]-[19] and the work shows that selection of appropriate strategies can be adaptively adjusted to solve constraint and multi-objective optimization problems. Trade-off involving self-adaptive measures lies in the limited number of fitness solutions that could be deployed, along with higher computation cost in terms of longer time consumption.

Another example of self-adaptive DE based on structural components can be found in [20]. The self-adaptive strategy are applied on F and CR but the setting of NP still depends on users. Besides that, another two parameters, τ_1 and τ_2 are introduced in [20] in the self-adaptive DE which require users to pre-determine the appropriate values. Additional two parameters τ_1 and τ_2 into existing number of parameters of DE increases the complexity of parameter tuning process. The work in [21] extends the adaptation scheme to adjust NP in DE. Based on the adaptation scheme, two DE models known as differential evolution in absolute encoding (DE-Abs) and DE-Rel are produced. For both DE-Abs and DE-Rel, users still need to determine the appropriate values of F and CR .

Ensemble-based DE could be reviewed as another dynamic attempt to accommodate the mutation and crossovers strategies in a more compact environment, whilst maintaining a steady convergence rate. Ensemble learning paradigm is seen applicable to rectify complex data series for data categories with wider degree of unpredictability and irregularity. Stated in [11], [16], [22]-[24], ensemble methods is beneficial in encouraging better convergence rate for larger population. Ensemble-based approach enables DE to determine its own parameter control therefore adjusting it to accommodate the most potential solution, usually within a predicted range. Mallipeddi and Suganthan also had stressed out the impact of population size, dimensionality, and mutation strategies in terms of leveraging the convergence rate and computational [24].

Based on the research that have been conducted so far,

we know that setting of F , CR and NP influences the generation of solution space, which eventually affects the convergence and robustness of DE in problem solving. A proper balance of parameter tuning between CR and F is required in the exploration and exploitation phases. During the evolution, parameter tuning can be performed as a counter strategy for error that might arise from the exploration of search space. Exploitation step can also administered in order to solve problems and obtaining solutions that would thrive longer. To address the problem, various DE's parameter selection methods are indeed suggested and the related work has been summarized in [2], [3]. Even though some ranges for the control parameters and their pre-conditions are suggested by the researchers, there is no clear and consistent justification for the setting. An adequate NP is needed to balance the trade-off between convergence and computation cost [13]. A large NP can extend the exploration space and increases the possibilities of finding good solutions. However, the evolution suffers from an undesirable and high computational cost. In contrast, an insufficient NP may cause premature convergence and local saturations.

The summarizations in [2], [3] also show that there is no adaptation strategy involving all control parameters and mutation strategy simultaneously. Besides that, the area is lack of research focusing on the adaptation scheme for either NP or mutation strategy, M .

Intervention from users to tune the control parameters during the evolution is required to balance the exploration and exploitation phases but the process requires a swift time consumption. However, selection of appropriate parameter values occasionally necessitate for a priori experience and problem dependent on user. Trade-off usually ensues between exploration and exploitation where selection of relevant control parameter and strategies remain ambiguous and arduous. Since both self-adaptive strategy and ensemble have shown promising result to set control parameters and mutation strategy, we would like to investigate their potential if they are integrated as a unifying scheme to adjust the control parameters as well as mutation strategy. In our research, we integrate the adaptation scheme from DE-Rel [21] to determine NP and ensemble from ESPDE [2] to determine CR , F and M in our DE model.

III. METHODOLOGY

We called our DE model as Self Adaptive Ensemble-based Differential Evolution or SAEDE owing to the use of self-adaptive and ensemble mechanisms to set the relevant parameters for each generation. Self-adaptation refers to evolution of solution encoded into chromosomes. On the other hand, an ensemble refers to a method constructing the final solution from a set of solutions based on consensus. In our model, self-adaptation is used to determine the value of NP while ensemble is used to determine the values of F , CR and M at each generation. The concept of using both self-adaptation and ensemble comes from two DE models known as DE-Rel [21] and ESPDE [2]. For DE-Rel, it can adaptively determine NP for each generation but the values of CR and F are held constant in the evolution. On the other hand, ESPDE uses ensemble to determine the

combination of CR , F and M in different stages of evolution but NP is held constant.

Similar to DE-Rel, the chromosome in SAEDE is represented by a parameter vector, $\mathbf{z}_{i,g} = \mathbf{x}_{i,g}, y_{i,g}$ as shown in Fig. 1, with $i = 1, 2, \dots, NP_g$. Here, NP_g refers to population size at generation- g and it varies across generations. $\mathbf{x}_{i,g}$ consists of a D -dimensional vector which represents the solution- i for an optimization problem at generation- g , i.e., $\mathbf{x}_{i,g} = \{x^1, \dots, x^D\}$. The variable $y_{i,g}$ refers to the population size's growth rate associated with individual- i at generation- g .

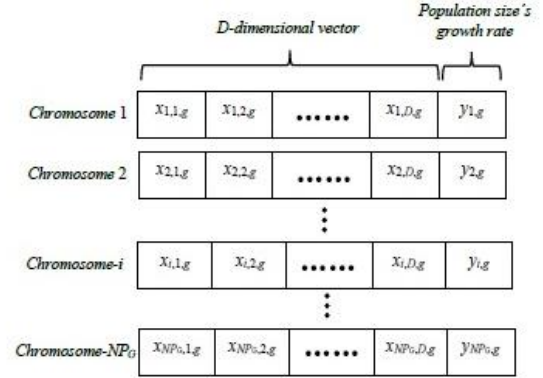


Fig. 1. Chromosome representations for SAEDE.

The initial j -th decision variable of the i -th solution at generation, $g = 0$ is generated within the search space constrained by the prescribed minimum and maximum decision variable's bounds $\mathbf{x}_{min} = \{x_{min}^1, \dots, x_{min}^D\}$ and $\mathbf{x}_{max} = \{x_{max}^1, \dots, x_{max}^D\}$ as shown in Equation 3. The initial NP_g at generation $g = 0$ is generated within the range of the prescribed minimum and maximum bounds of $[10D, 100D]$.

After initialization, the population of $y_{i,g}$ of generation- g determines the population size of next generation, NP_{g+1} . The population size's growth rate for generation- $g + 1$, y_{g+1} is determined by taking the average of the population $y_{i,g}$ as shown in Equation 1. Then, y_{g+1} is used to calculate the value of NP_{g+1} for SAEDE based on Equation 1. If the NP_{g+1} is larger than NP_g , a number of solutions are randomly generated within their respective bounds to fulfill population size NP_{g+1} . However, if NP_{g+1} is smaller than NP_g , a number of least fit solutions are deleted to meet population size NP_{g+1} .

$$y_{g+1} = (int) \left(\frac{y_{1,g} + y_{2,g} + \dots + y_{NP_g,g}}{y_g} + 0.5 \right) \quad (1)$$

$$NP_{g+1} = (int)(y_{g+1} * NP_g) + NP_g + 0.5 \quad (2)$$

$$\mathbf{x}_{i,0} = x_{min}^j + rand(0,1) * (x_{max}^j - x_{min}^j) \quad j = 1, 2, \dots, D \quad (3)$$

Similar to original DE, the population of individuals in SAEDE evolves over generations through mutation, crossover and selection. However, SAEDE does not use constant CR , F and M as commonly practiced in original DE. Instead, SAEDE dynamically changes the control parameters and mutation strategy as demonstrated in ESPDE. The term of control parameters refers to CR and F only.

For EPSDE, CR is taken in the range 0.1-0.9 in steps of 0.1

while F is taken in the range 0.4-0.9 in steps of 0.1 as suggested in [2]. Since there are 9 and 5 options for the setting of CR and F respectively, the total combinations of CR and F are 45 (9×5). For crossover schemes, EPSDE has 3 mutation strategies as suggested in [2]. The same mutation strategies are included in SAEDE, i.e., {DE/rand/1/bin, DE/best/2/bin, DE/current-to-rand/1/bin}. The options of crossover schemes are shown in Table I. Therefore, EPSDE contains 135 (45×3) configurations of mutation strategy and control parameters in the pool. Setting of the mutation strategy and control parameters across the generations are chosen from the pool of 135 configurations.

TABLE I: SETTINGS OF CONTROL PARAMETERS AND MUTATION STRATEGIES FOR DIFFERENT DES

Settings	SAEDE	EPSDE	DE-Rel
NP	[10D, 100D]	10D	[10D, 100D]
CR	[0.1:0.1:0.9]	[0.1:0.1:0.9]	0.5
F	[0.4:0.1:0.9]	[0.4:0.1:0.9]	0.5
strategy	{DE/rand/1/bin, DE/best/2/bin, DE/current-to-rand/1/bin}	{DE/rand/1/bin, DE/best/2/bin, DE/current-to-rand/1/bin}	DE/rand/1/bin

Initially, each target vector $\mathbf{z}_{i,g}$ in the initial population is randomly assigned with a configuration of mutation strategy and control parameters. This means the mutation and crossover operations experienced by the target vectors depend on the assigned configurations. The CR , F and mutation strategy associated with the i -solution at generation- g are denoted as $CR_{i,g}$, $F_{i,g}$ and $M_{i,g}$ respectively. At generation- g , the individuals in current population, P_z are known as target vector and they are correspond to $\mathbf{x}_{i,g}$, $y_{i,g}$. Corresponding to each target vector $\mathbf{x}_{i,g}$, a mutant vector, $\mathbf{v}_{i,g} = \{v^1, \dots, v^D\}$ is produced through the differential mutation operation. The same differential mutation operation is also extended to the last gene in each chromosome corresponding to $y_{i,g}$ to produce a mutant scalar, $\dot{y}_{i,g}$. The differential mutation operation adds a scaled, randomly sampled, vector difference to a third vector as shown in Equation 4 to Equation 6. At this stage, the population is called intermediary population, $P_z = \dot{\mathbf{z}}_{i,g} = \{\mathbf{v}_{i,g}, \dot{y}_{i,g}\}$.

The associated $F_{i,g}$ is used in the differential mutation operation to control the rate at which $\mathbf{z}_{i,g}$ evolves. Three mutation strategies used in SAEDE are shown in Equation 4 to Equation 6. Each vector in the current population, P_z is then combined with a mutant vector based on the setting of $CR_{i,g}$ to produce trial vector $\mathbf{u}_{i,g} = \{u^1, \dots, u^D\}$ based on Equation 7 and trial scalar $\dot{y}_{i,g}$ based on Equation 8. As a result, a trial population $P_z = \dot{\mathbf{z}}_{i,g} = \{\mathbf{u}_{i,g}, \dot{y}_{i,g}\}$ is produced. During mutation and crossover operations, if any decision variables value of newly generated trial vectors exceeds their corresponding upper or lower bounds, then they are re-initialized randomly and uniformly within the pre-specified range until they fall within the bounds.

$$\text{DE/best/2 [25]: } \dot{\mathbf{z}}_{i,g} = \mathbf{z}_{best,g} + F(\mathbf{z}_{r1,g}^i - \mathbf{z}_{r2,g}^i) + F(\mathbf{z}_{r3,g}^i - \mathbf{z}_{r4,g}^i) \quad (4)$$

$$\text{DE/rand/1 [25]: } \dot{\mathbf{z}}_{i,g} = \mathbf{z}_{r1,g}^i + F(\mathbf{z}_{r2,g}^i - \mathbf{z}_{r3,g}^i) \quad (5)$$

$$\text{DE/current - rand/1 [26]: } \dot{\mathbf{z}}_{i,g} = \mathbf{z}_{r1,g}^i + K(\mathbf{z}_{r1,g}^i - \mathbf{z}_{r2,g}^i) + F(\mathbf{z}_{r3,g}^i - \mathbf{z}_{r4,g}^i) \quad (6)$$

$$u_{i,g}^j = \begin{cases} v_{i,g}^j, & \text{if } (rand_j[0,1] \leq CR_{i,g}) \\ & \text{or } (j = j_{rand}) \\ & j = 1, 2, \dots, D \\ x_{i,g}^j, & \text{otherwise} \end{cases} \quad (7)$$

$$\dot{y}_{i,g} = \begin{cases} \dot{y}_{i,g}, & \text{if } (rand_j[0,1] \leq CR_{i,g}) \quad j = D + 1 \\ y_{i,g}, & \text{otherwise} \end{cases} \quad (8)$$

During selection, the objective function of all trial vectors $\mathbf{u}_{i,g}$ are evaluated. In a minimization problem, if the objective function of a trial vector, $f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g})$, it replaces the target vector in the next generation; otherwise, the target vector retains its place in the population as shown in Equation 9 to Equation 10. At the same time, the configurations associated in the production of trial vectors are evaluated. If the generated trial vector produce better objective function than its target vector, then the associated configuration of mutation strategy and control parameters will be retained in the next generation and store in a successful archive, A . Otherwise, if the target vector has better objective function than its trial vector, the associated configuration of mutation strategy and control parameters will be randomly re-initialized with a new configuration or randomly chosen from the available configurations from the successful archive. The genetic operations of mutation, crossover and selection are repeated generation after generation until stopping criterion is satisfied.

Therefore, SAEDE combines the autonomous procedures of setting control parameters, mutations strategies from EPSDE and population size from DE-Rel. This means that the values of F , CR , NP and M will be automatically adjusted across the generation with minimum intervention from users.

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g}, & \text{if } f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g}, & \text{otherwise} \end{cases} \quad (9)$$

$$y_{i,g+1} = \begin{cases} \dot{y}_{i,g}, & \text{if } f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g}) \\ y_{i,g}, & \text{otherwise} \end{cases} \quad (10)$$

The pseudocode of our proposed model, SAEDE is summarized in Algorithm 1.

Algorithm 1: Pseudocode for SAEDE

Step 1: Initialisation

- Step 1.1 Set $g = 0$. Randomly initialize population size, NP_g .
- Step 1.2 Randomly initialize a population of NP_g individuals, $P_z = \{\mathbf{z}_{i,g}, \dots, \mathbf{z}_{NP_g,g}\}$.
- Step 1.3 Initialise a pool of configurations of control parameters and mutation strategies.

Step 2: Randomly assign each population individual with a configuration of control parameters and mutation strategies.

Step 3: Evolution

while stopping criterion is not met **do**

Step 3.1 Mutation Step

for $i = 1$ to N_g **do**

Generate a mutated vector $\mathbf{v}_{i,g} = \{v_{i,g}^1, v_{i,g}^2, \dots, v_{i,g}^D\}$ and mutated scalar $\dot{y}_{i,g}$ corresponding to target vector $\mathbf{z}_{i,g}$ by using the

configuration of control parameters and mutation strategy associated with the target vector

end for

Step 3.2: Crossover

for $i = 1$ to N_g **do**

$j_{\text{rand}} = \text{rand}[0, 1] \cdot D$

for $j = 1$ to D **do**

Generate a trial vector $\mathbf{u}_{i,g} = \{u_{i,g}^1, u_{i,g}^2, \dots, u_{i,g}^D\}$ for each target vector $\mathbf{x}_{i,g}$ associated with $\mathbf{z}_{i,g}$ based on Equation 9.

end for

for $j = D+1$ **do**

Generate a trial scalar $\dot{y}_{i,g}$ for each target scalar $y_{i,g}$ associated with $\mathbf{z}_{i,g}$ based on Equation 10.

end for

end for

Step 3.3: Selection

for $i = 1$ to N_g **do**

if $f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g})$ **then**

$\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}, y_{i,g+1} = \dot{y}_{i,g}$

$f(\mathbf{x}_{i,g+1}) \leq f(\mathbf{u}_{i,g})$

Copy the configuration of control parameters and mutation strategy associated with the trial vector to the successful archive, A .

if $f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{\text{best},g})$ **then**

$\mathbf{x}_{\text{best},g} = \mathbf{u}_{i,g}, y_{\text{best},g} = \dot{y}_{i,g}$

$f(\mathbf{x}_{\text{best},g}) \leq f(\mathbf{u}_{i,g})$

end if

else

$\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$

$y_{i,g+1} = y_{i,g}$

end if

end for

Step 3.4: Update configurations

for $i = 1$ to N_g **do**

if $f(\mathbf{u}_{i,g}) > f(\mathbf{x}_{i,g})$ **then**

Randomly select a new configuration of control parameters and mutation strategy from the pool or from the successful archive.

end if

end for

Step 3.5: Population size adaptation

for $i = 1$ to N_g **do**

Calculate y_{g+1} based on Equation 1

Calculate NP_{g+1} based on Equation 2

end for

end while

[21], [27].

The evolutions for all three models of DEs run 100,000 generations ($G = 100,000$) and are repeated by using k -th seed numbers, $k = 1, 2, \dots, K$. The same 30 seed numbers are used to evaluate different DEs, $K = 30$. The evolutionary processes are terminated if the best-fitness, $f_{\text{best}} < 1 \times 10^{-20}$, otherwise the processes continue until they reach G .

In the investigation of parameter setting, the performances of different models of DEs are compared. Twenty benchmark functions in [21] are adopted to evaluate the DEs. A summary about the 20 benchmark function are shown in Table II. All functions are minimization of problems with their best minimum solution denoted as f_{min} . The number of variables, D involved varies from 2 to 10 and the ranges of variable search is represented by S . However, the work [28] has shown that the final results obtained by DEs might have a certain residual of error rather than zero due to the precision problem of software. This problem happens to function F9 in our experiments. When the actual optimal values are substituted into function F9, the value of f_{min} is 8.88×10^{-16} but not identical zero. This problem be taken into consideration into the discussion of results later.

V. RESULT AND ANALYSIS

To investigate the performances of different DEs, several measurements are used in the investigation such as the average of best-fitness, f_{best} , success rate (SR), hypothesis test and analysis of bias-variance. Details of some of the measurements are explained in the section later.

When the $f_{\text{best}} < 1 \times 10^{-20}$, the evolution for this run is called a successful run. SR computes the number of successful runs against the number of independent runs. Therefore SR actually measure the success probability of DEs. The performances of DEs based on SR and f_{best} are shown in Table III. Based on the results, SAEDE achieved the highest but not full SR in 1 case and full SR in 16 cases. Therefore, SAEDE is associated the best SR in 17/20 cases in total. On the other hand, both ESPDE and DE-Rel have achieved full SR in the same 15 cases respectively.

All of the DE models has no successful runs at all in 3 functions, i.e., F9, F13 and F17. However, the unsuccessful runs in F9 may be caused by the software's precision problem which has been demonstrated in [28]. The value of f_{min} for F9 is not able to achieve zero but approximately 8.88×10^{-16} even though the actual optimal values are substituted into the function. The difference between the f_{best} obtained by all models and the actual f_{min} is very small, i.e., 1.12×10^{-16} . For the 2 cases in which all DEs do not achieve full SR , analysis bias-variance are carried out. Bias-variance analysis is performed to evaluate how accurate and how consistent the prediction of the algorithms as compared to the actual global optimal solution. Since we do know the actual global optimal solution for the benchmark functions, such analysis can be carried out. However, the analysis of bias-variance is carried out to evaluate how accurate and how consistent of the unsuccessful runs' solutions from the actual global points for DEs which do not achieve full SR . For the explanation on bias-variance, assumes that a single estimator after learning produces estimates $p(\mathbf{x})$, so the squared error between the estimates $p(\mathbf{x})$ and target function ($t(\mathbf{x})$).

IV. EXPERIMENTAL SETUP

We would like to stress that the comparison of different DE models are performed under similar conditions. More heuristic strategies may give different results. Therefore, the similar experimental setups used by DE-Rel in [21] and EPSDE in [2] are considered in the implementation of DEs in our experiments. This is to minimize the nuisance factors affecting the results and enable the same analyses to be carried out for them. For ease of understanding, the settings of the mutation strategies and control parameters for the DE models are summarized in Table I. For DE-Rel, the DE/rand/1/bin strategy is chosen for the crossover process because it is the first and also the most widely used scheme

$$\{p(\mathbf{x}) - \langle t|\mathbf{x} \rangle\}^2 \quad (11)$$

The squared error depends on the types of estimators, the setting of estimator's control parameters and the training data to train estimator. The expected error produced by the model can be written as:

$$E\{p(\mathbf{x}) - \langle t|\mathbf{x} \rangle\}^2 \quad (12)$$

Using some algebraic manipulation [29], equation (12) can be rewritten as:

$$E\{p(\mathbf{x}) - \langle t|\mathbf{x} \rangle\}^2 = \{E[p(\mathbf{x})] - \langle t|\mathbf{x} \rangle\}^2 + E\{p(\mathbf{x}) - E[p(\mathbf{x})]\}^2 \quad (13)$$

The first term refers to the squared bias, and the second term refers to the variance. Bias measures the difference between the average estimate over all possible samples and the true function. On the other hand, the variance measures the difference between an estimate obtained for a sample and the average estimate obtained over all possible samples. The estimates from samples could be different due to training data and the setting of estimator model. Table IV shows the analysis of bias-variance on the best obtained solution for the DE models, where bias is denoted as β and variance are denoted as ς .

TABLE II: DETAILS ON THE BENCHMARK FUNCTIONS

Denotation	Test Function	L	S	f_{\min}	X^*	U/M
F1	De Jong's function 1 (Sphere Model)	10	$[-100, -90]^n$	0	$(0, 0, \dots, 0)$	U
F2	De Jong's function 2 (Rosenbrock's Saddle)	2	$[-2.048, 2.048]^n$	0	$(1, 1)$	U
F3	Camel back – three hump problem	2	$[-5.0, -4.5]^n$	0	$(0, 0)$	M
F4	Becker and Lago's problem	2	$[-10, -9]^n$	0	$(\pm 5, 5)$	M
F5	Schwefel's problem 2.22	10	$[-10, -9]^n$	0	$(0, 0, \dots, 0)$	U
F6	Rastrigin's function	10	$[-5.120, -4.608]^n$	0	$(0, 0, \dots, 0)$	M
F7	Modified Rosenbrock's problem	2	$[-5.0, -4.5]^n$	0	$(1, 1), (0.3412, 0.1164)$	M
F8	Briewangk's problem	10	$[-600, -540]^n$	0	$(0, 0, \dots, 0)$	M
F9	Ackley's path function	10	$[-32.0, -28.8]^n$	0	$(0, 0, \dots, 0)$	M
F10	Bohachevsky's problem 2	2	$[-50, -48]^n$	0	$(0, 0)$	M
F11	Rotate hyper-ellipsoid function	10	$[-65.5360, -58.9824]^n$	0	$(0, 0, \dots, 0)$	U
F12	Sum of different power problem	10	$[-1.0, -0.9]^n$	0	$(0, 0, \dots, 0)$	U
F13	Miele and Cantrell's problem	4	$[-1.0, -0.9]^n$	0	$(0, 1, 1, 1)$	M
F14	Schaffer's problem 1	2	$[-100, -90]^n$	0	$(0, 0)$	M
F15	Moved axis parallel hyper-ellipsoid function	10	$[-5.120, -4.608]^n$	0	$(5*N, 5*N, \dots, 5*N)$	U
F16	Helical valley problem	3	$[-10, -9]^n$	0	$(1, 0, 0)$	U
F17	Salomon's problem	10	$[-100, -90]^n$	0	$(0, 0, \dots, 0)$	M
F18	Powell's quadratic problem	4	$[-10, -9]^n$	0	$(0, 0, 0, 0)$	U
F19	Bohachevsky's problem 1	2	$[-50, -45]^n$	0	$(0, 0)$	M
F20	Wood's function	4	$[-10, -9]^n$	0	$(1, 1, 1, 1)$	U

TABLE III: SR AND f_{best} FOR DIFFERENT DE MODELS

Function	SAEDE		ESPDE		DE-Rel	
	SR	f_{best}	SR	f_{best}	SR	f_{best}
F1	1	8.2E-21	1	7.8E-21	1	7.9E-21
F2	1	5.5E-21	1	4.5E-21	1	6.7E-21
F3	1	4.5E-21	1	3.8E-21	1	3.5E-21
F4	1	4.3E-21	1	4.3E-21	1	4.3E-21
F5	1	9.2E-21	1	8.7E-21	1	8.7E-21
F6	1	0.00	1	0.00	1	0.00
F7	0.8	1.7E-03	0.5	3.7E-03	0.7	2.2E-03
F8	1	0.00	1	0.00	1	0.00
F9	0	8.9E-16	0	8.9E-16	0	8.9E-16
F10	1	0.00	1	0.00	1	0.00
F11	1	8.4E-21	1	7.4E-21	1	7.9E-21
F12	1	6.7E-21	1	6.8E-21	1	6.4E-21
F13	0	1.0E-08	0	1.0E-08	0	1.0E-08
F14	1	0.00	0.9	6.5E-04	0.9	9.7E-04
F15	1	7.8E-21	1	8.3E-21	1	7.9E-21
F16	1	6.0E-21	1	5.9E-21	1	4.9E-21
F17	0	1.0E-01	0	1.0E-01	0	1.0E-01
F18	1	5.7E-21	1	5.7E-21	1	6.2E-21
F19	1	0.00	1	0.00	1	0.00
F20	1	6.7E-21	1	6.3E-21	1	6.5E-21

For function F7, the values of β for all DEs are the same. However, DE-Rel has the lowest variance followed by SAEDE and ESPDE. This means all DEs have the same level of accuracy but differs in terms of consistency for

function F7. DE-Rel is the most consistent followed by SAEDE and ESPDE in function F7. For function F14, SAEDE has no unsuccessful runs but ESPDE and DE-Rel each has 15 and 9 unsuccessful runs respectively. Both ESPDE and DE-Rel has $\varsigma = 0$ which means they consistently produce the same value of f_{best} for a number of repetitions. However, their associated f_{best} varies from the actual global optimal point by $9.72\text{E-}03 (\sqrt{9.44\text{E} - 05})$ in average as compared to SAEDE achieving $f_{\text{best}} < 10\text{E-}20$. This means SAEDE produces much better f_{best} than ESPDE and DE-Rel for function F14.

For the 15 cases in which all DE models achieved full SR, hypothesis tests are conducted to compare whether our proposed SAEDE is significantly better than ESPDE and DE-Rel or not. A pairwise t -test is carried out at the significance level of 0.05 to evaluate the significance differences for f_{best} between the SAEDE–ESPDE and SAEDE–DE-Rel. Therefore, the hypothesis is to evaluate the null hypothesis that f_{best} of SAEDE is significantly larger than the other two models, against the alternative hypothesis that f_{best} of SAEDE is not larger than the other two models. The results of hypothesis tests for the 15 cases are shown in Table V. The hypothesis tests show that ESPDE and DE-Rel each has one case associated with significantly better f_{best} than SAEDE respectively. For functions F6, F8, F10, and F19, their p -values are “NaN” and it means there is no variance between the samples of f_{best} for the paired DEs. In

other words, the samples for the paired DEs are exactly the same in 4 functions. Out of 20 cases, the performances of

our proposed SAEDE are as compatible as EPSDE and DE-Rel in 15 cases whereby all of them achieve $f_{best} < 1 \times 10^{-20}$.

TABLE IV: BIAS-VARIANCE FOR DIFFERENT DES

Function	SAEDE		ESPDE		DE-Rel	
	β	ς	β	ς	β	ς
F7	5.50E-05	3.01E-36	5.50E-05	6.52E-36	5.50E-05	2.01E-36
F14	-	-	9.44E-05	0.00	9.44E-05	0.00

TABLE V: PAIRWISE T-TEST FOR SAEDE-ESPDE AND SAEDE-DE-REL FOR THE SELECTED 15 CASES

Function	SAEDE-ESPDE		SAEDE-DE-Rel	
	p-value	Reject H_0	p-value	Reject H_0
F1	0.19	0	0.24	0
F2	0.10	0	0.95	0
F3	0.20	0	0.14	0
F4	0.52	0	0.92	0
F5	0.06	0	0.02	1
F6	NaN	-	NaN	-
F8	NaN	-	NaN	-
F10	NaN	-	NaN	-
F11	0.01	1	0.10	0
F12	0.54	0	0.33	0
F15	0.88	0	0.59	0
F16	0.43	0	0.06	0
F18	0.46	0	0.73	0
F19	NaN	-	NaN	-
F20	0.27	0	0.32	0

Overall, our proposed SAEDE has better SR than ESPDE and DE-Rel. As for the measurement based on \bar{f}_{best} , the three DE models are quite compatible and it is supported by the hypothesis tests. The main advantage of SAEDE as compared to ESPDE and DE-Rel is it does not require explicit setting of NP , CR , F and M .

VI. CONCLUSION AND FUTURE WORK

In our work we have combined the self-adaptive scheme from DE-Rel and the ensemble scheme from EPSDE to develop SAEDE. The performances of these models are compared. The results show that SAEDE is slightly better in SR than the other two DE models and their f_{best} are compatible. The main advantage of SAEDE over the other DEs is it able to determine the appropriate population size, control parameters and mutation strategy to solve different benchmark functions with minimum intervention from a user. SAEDE uses different configurations of control and mutation strategy in different stages of evolution to ensure adequate performances. In consideration of the possible future work, we would like to explore other possible self-adaptive schemes in DE and investigates their effects in performances specifically in problem complexity. Besides that, we could include other mutation strategies into the development of SAEDE.

ACKNOWLEDGMENT

This work is funded by the Ministry of Higher Education Malaysia's Fundamental Research Grant Scheme (FRGS) with grant number 203/PCGSS/6711510 and FRGS/1/2017/ICT02/UPSI/02/3.

REFERENCES

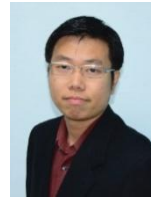
- [1] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [2] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.
- [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] L. Titare, P. Singh, L. Arya, and S. Choubey, "Optimal reactive power rescheduling based on EPSDE algorithm to enhance static voltage stability," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 588–599, 2014.
- [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] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
- [7] A. Zamuda and J. Brest, "Self-adaptive control parameters randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.
- [8] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, no. 1–2, pp. 61–106, Feb. 2010.
- [9] Y. Zhou, J. Wang, Y. Zhou, Z. Qiu, Z. Bi, and Y. Cai, "Differential evolution with guiding archive for global numerical optimization," *Applied Soft Computing*, vol. 43, pp. 424–440, 2016.
- [10] D. Zaharie, "Control of population diversity and adaptation in differential evolution algorithms," in *Proc. MENDEL*, vol. 9, 2003, pp. 41–46.
- [11] M. Niu, Y. Wang, S. Sun, and Y. Li, "A novel hybrid decomposition-and-ensemble model based on ceemd and gwo for short-term pm 2.5 concentration forecasting," *Atmospheric Environment*, vol. 134, pp. 168–180, 2016.
- [12] S. M. Mohseni-Bonab, A. Rabiee, B. Mohammadi-Ivatloo, S. Jalilzadeh, and S. Nojavan, "A two-point estimate method for uncertainty modeling in multi-objective optimal reactive power dispatch problem," *International Journal of Electrical Power & Energy Systems*, vol. 75, pp. 194–204, 2016.
- [13] W. Zhu, Y. Tang, J.-A. Fang, and W. Zhang, "Adaptive population tuning scheme for differential evolution," *Information Sciences*, vol. 223, pp. 164–191, 2013.
- [14] T. Krink, B. Filipic, and G. B. Fogel, "Noisy optimization problems—a particular challenge for differential evolution?" in *Proc. IEEE Congress on Evolutionary Computation, CEC2004*, vol. 1, 2004, pp. 332–339.
- [15] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Proc. the 2005 IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 506–513.
- [16] R. Mallipeddi, G. Iacca, P. N. Suganthan, F. Neri, and E. Mininno, "Ensemble strategies in compact differential evolution," in *Proc. 2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 1972–1977.
- [17] M. Asafuddoula, T. Ray, and R. Sarker, "A differential evolution algorithm with constraint sequencing: An efficient approach for problems with inequality constraints," *Applied Soft Computing*, vol. 36, pp. 101–113, 2015.
- [18] Z. Zhao, J. Yang, Z. Hu, and H. Che, "A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric latin hypercube design for unconstrained optimization problems," *European Journal of Operational Research*, vol. 250, no. 1, pp. 30–45, 2016.
- [19] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Computers & Operations Research*, vol. 67, pp. 155–173, 2016.
- [20] J. Brest, V. Zumer, and M. Maucec, "Self-adaptive differential evolution algorithm in constrained real-parameter optimization," in *Proc. IEEE Congress on Evolutionary Computation, CEC 2006*, 2006, pp. 215–222.

- [21] N. S. Teng, J. Teo, and M. H. A. Hijazi, "Self-adaptive population sizing for a tune-free differential evolution," *Soft Computing*, vol. 13, no. 7, pp. 709-724, 2009.
- [22] Y. Xue, S. Zhong, Y. Zhuang, and B. Xu, "An ensemble algorithm with self-adaptive learning techniques for high-dimensional numerical optimization," *Applied Mathematics and Computation*, vol. 231, pp. 329-346, 2014.
- [23] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329-345, 2016.
- [24] R. Mallipeddi and P. N. Suganthan, "Ensemble differential evolution algorithm for cec2011 problems," in *Proc. 2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 1557-1564.
- [25] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.
- [26] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *Proc. Australasian Joint Conference on Artificial Intelligence*. Springer, 2004, pp. 861-872.
- [27] 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.
- [28] 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.
- [29] M. Keijzer and V. Babovic, "Genetic programming, ensemble methods and the bias/variance tradeoff-introductory investigations," in *Proc. Genetic Programming*, ser. *Lecture Notes in Computer Science*, vol. 1802, Springer Berlin / Heidelberg, 2000, pp. 76-90.



Shir Li Wang was born on July 14, 1979 in Selangor, Malaysia. She received the B. Tech. (Hons.) in quality control & instrumentation, in 2002 and M.Sc. in electrical & electronic engineering, in 2007, both from the Universiti Sains Malaysia, Penang, Malaysia. She obtained her Ph.D. degree from the University of New South Wales, NSW, Australia, in 2012. She is currently a senior lecturer at the Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris. Her research interests

include machine learning, data mining, optimization, and adversarial learning. She is a Member of the IEEE since 2013.



Theam Foo Ng was born on July 25, 1979 in Perak, Malaysia. He received the B.Sc. (Hons.) in mathematics, in 2002 and M.Sc. in statistics, in 2006, both from the Universiti Sains Malaysia (USM), Penang, Malaysia. He obtained his Ph.D. degree from the University of New South Wales, NSW, Australia, in 2012. He is currently a senior lecturer at Centre for Global Sustainability Studies, USM. His research interests include machine learning, pattern recognition, computational intelligence, image processing and educational for sustainable development. He is a Member of the IEEE since 2011.



Farid bin Morsidi was born in Sarawak, Malaysia to multiracial background in July 18, 1990. He is multilingual with 5 strong local languages preferably Mandarin & English. He is a candidate of final year with masters degree in information systems at Universiti Pendidikan Sultan Idris. He is passionate in research areas involving data clustering & classifications, fuzzy algorithms, & evolutionary computing. His recent projects include named entity recognition in malay & self-adaptive in differential evolution. He is working as an research assistant at Computing Department for 3 years. He had published 2 papers relating in Named Entity Recognition.