Intelligent Automation And Soft Computing, 2020
Vol. 26, no. 3, 407–420
DOI: 10.32604/iasc.2020.013917

Optimized PID Controller using Adaptive Differential Evolution with Mean-of-*p*best Mutation Strategy

Ti-Hung Chen[1] and Ming-Feng Yeh[2,*]

[1]Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan, Taiwan, ROC
[2]Department of Electrical Engineering, Lunghwa University of Science and Technology, Taoyuan, Taiwan, ROC.

ABSTRACT
On the basis of JADE (adaptive differential evolution with optional external archive) and the modified differential evolution with *p*-best crossover (MDE_*p*BX), this study attempts to propose a modified mutation strategy termed "DE/$(\overline{p\text{best}})$/1" for the differential evolution (DE) algorithm, where "$(\overline{p\text{best}})$" represents the mean of p top-best vectors. Two modified parameter adaptation mechanisms are also proposed to update the crossover rate and the scale factor, respectively, in an adaptive manner. The DE variant with the proposed mutation strategy and two modified adaptation mechanisms is termed adaptive differential evolution with mean-of-*p*best mutation strategy, denoted by ADE_$\bar{p}$BM. In this study, the proposed two schemes are used not only to preserve the diversity of the population and prevent the premature convergence but also improve the search performance. The results of global optimization problems and the PID controller designs show that ADE_$\bar{p}$BM is comparable to or better than the four state-of-the-art adaptive DE variants in terms of accuracy, reliability and efficiency.

KEY WORDS: Differential evolution, mutation strategy, parameter adaptation scheme, PID control

1 INTRODUCTION

THE differential evolution (DE) algorithm is a population-based stochastic search technique (Storn & Price (1997)) and has been shown to be a simple yet powerful evolutionary algorithm for many real-world optimization problems, such as; function optimization, pattern recognition, power dispatch, antenna design, chemical processes and control systems (Xue et. al. (2015); Zhang &Sanderson (2009); Das and Suganthan (2011); Li et. al. (2013); Lu et. al. (2014)). In the DE algorithm, the five widely-used mutation strategies are "DE/rand/1", "DE/best/1", "DE/rand/2", "DE/best/2", and "DE/current-to-best/1" (Zhang & Sanderson (2009); Das & Suganthan (2011); Lu et. al. (2014)). The target vectors generated by a greedy mutation strategy, such as; "DE/best/1", "DE/current-to-best/1", and "DE/best/2", are generally attracted by the same best vector found so far by the entire population. The fact may lead to problems such as; premature convergence due to the resultant reduced population diversity, especially when solving multimodal problems. Two less greedy and more explorative variants of the "DE/current-to-best/1" mutation strategy, "DE/current-to-*p*best/1" (Zhang & Sanderson (2009)) and "DE/current-to-*gr*_best/1" (Islam et. al. (2012)), were proposed to overcome the premature convergence, where *p*best represents a randomly selected vector from the *p* top-ranked individuals at the current generation and *gr*_best means the best of the *q*% vectors randomly chosen from the current population. The central idea of these two strategies is to utilize the best or near-best member selected from a dynamically small pool to perturb the target vector. Such a scheme could preserve the diversity of the population so that the problem of premature convergence can be alleviated. Inspired by the above two mutation variants, this study attempts to develop another kind of less greedy and more explorative mutation strategy. Different from them, the proposed scheme substitutes the mean of the *p* top-best individuals $(\overline{p\text{best}})$ for the best vector in "DE/best/1". Such a mutation strategy is termed "DE/ $\overline{p\text{best}}$ /1" in this study. Analogously to

CONTACT Ming-Feng Yeh mfyeh@mail.lhu.edu.tw

"DE/current-to-$p$best/1" and "DE/current-to-gr_best/1", the target solutions generated by the proposed mutation strategy are not always attracted towards the same best vector found so far by the entire population, and this feature is helpful in avoiding a premature convergence at a local optimum.

The performance of the DE algorithm is also sensitive to the settings of the control parameters (e.g., the scale factor F and the crossover rate Cr). The best settings of the control parameters may be different for different problems. To successfully solve a specific optimization problem, it is generally necessary to fine-tune the control parameters by a time-consuming trial-and-error procedure (Qin et. al. (2009)). To solve this problem and make the performance of the DE more robust, several adaptive or self-adaptive mechanisms have been proposed to automatically find the proper trial vector generation strategies or suitable parameter settings during the search process (Zhang & Sanderson (2009); Das & Suganthan (2011); Lu et. al. (2014); Abbasa et. al. (2018)). If well designed, the strategy or parameter adaptation can improve the search performance and the robustness of an algorithm. JADE (Zhang & Sanderson (2009)), self-adaptive differential evolution (SaDE) Qin et. al. (2009)), modified DE with p-best crossover (MDE_pBX) (Islam et. al. (2012)), jDE (Brest et. al. (2006)) and the ensemble of control parameters and mutation strategies with DE (EPSDE) (Mallipeddi et. al. (2011)) are well-known adaptive DE (ADE) variants. Empirical studies have shown that the parameter adaptation technique can lead to these ADE variants with superior performance. Among the previous ADE variants, both JADE and MDE_pBX generate new F values according to a truncated Cauchy distribution with a scale parameter γ_F of 0.1 and new Cr values according to a normal distribution with standard deviation σ_{Cr} of 0.1. In addition, the location parameter of the Cauchy distribution (μ_F) and the mean of normal distribution (μ_{Cr}) are updated using new successful F and Cr values, respectively, at each generation. The main difference between them is that the updating rules of μ_F and μ_{Cr} for JADE and MDE_pBX are in different manners discussed later in the next section. Note that both the scale parameter γ_F (JADE) and the standard deviation σ_{Cr} (MDE_pBX) remain constant during the search process. This gives rise to propose the motivation to a novel self-adaptation scheme that can simultaneously adapt μ_F and μ_{Cr} as well as γ_F and σ_{Cr} during the search process. The central idea of the proposed adaptation scheme is also inherited from JADE and MDE_pBX. That is, new F values are generated according to a truncated Cauchy distribution and new Cr values according to a normal distribution. However, in this study, μ_F and γ_F are updated using new successful F values at each generation, while μ_{Cr} and σ_{Cr} using new successful Cr values. Such a modification attempts to further improve the robustness and convergence performance of DE algorithm.

Integrating the proposed mutation strategy "DE/$\overline{p\text{best}}$/1" with the aforementioned parameter adaptation scheme forms a new ADE variant termed ADE_ $\bar{p}$ BM (ADE with mean-of-$p$best mutation strategy) hereafter. In order to demonstrate the search effectiveness, the developed ADE_$\bar{p}$BM algorithm is compared with four state-of-the-art ADE variants over a set of 12 benchmark functions on real parameter optimization.

Owing to the proportional-integral-derivative (PID) controllers with the advantage of a simple structure, good stability, and high reliability, they are still widely applied in the industrial processes now (Tabatabaei & Barati-Boldaji (2017); Kim et. al. (2018)). This study therefore focuses on the optimization of the PID controller system by using the DE algorithm. Three PID control gains are; proportional gain K_P, integral gain K_I and derivative gain K_D , which are determined by the proposed ADE_$\bar{p}$BM and the four ADE variants such that the pre-defined objective function is minimized.

The remainder of this study is organized as follows: Section 2 briefly represents some background material of DE algorithms and PID controller design, the proposed mutation strategy "DE/$\overline{p\text{best}}$/1". The corresponding parameter adaptation scheme is described in Section 3. Section 4 represents the search performance of the proposed algorithm for 12 benchmark functions and two PID controller design problems. Section 5 concludes this study.

2 PRELIMINARIES

2.1 *Differential Evolution Algorithm*

ASSUME that a population contains N_p individuals and each individual is in the form of a D-dimensional vector as; $\mathbf{x}_{i,G} = (x_{1i,G}, x_{2i,G}, \ldots, x_{Di,G})$, where G denotes at the generation G and i = 1, 2, …, N_p. Note that an individual (target vector) represents a potential solution of the optimization problem. The DE algorithm begins with a randomly generated population within the search space. After initialization, the DE iteratively uses the trial vector generation strategy (i.e., mutation and crossover operations) and the selection operation to evolve the population until a stopping criterion is met.

Mutation: The following are five most frequently used mutation strategies for generating a mutant vector $\mathbf{v}_{i,G}$ (Zhang & Sanderson (2009); Das & Suganthan (2011); Lu et. al. (2014)):

"DE/rand/1":

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}), \tag{1}$$

"DE/best/1":

$$\mathbf{v}_{i,G} = \mathbf{x}_{best,G} + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}), \tag{2}$$

"DE/current-to-best/1"
("DE/target-to-best/1" or "DE/rand-to-best/1"):

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F(\mathbf{x}_{best,G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}), \quad (3)$$

"DE/rand/2":

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}) + F(\mathbf{x}_{r4,G} - \mathbf{x}_{r5,G}), \quad (4)$$

"DE/best/2":

$$\mathbf{v}_{i,G} = \mathbf{x}_{best,G} + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) + F(\mathbf{x}_{r4,G} - \mathbf{x}_{r5,G}), \quad (5)$$

where the indices $r1$, $r2$, $r3$, $r4$, and $r5$ are distinct integers randomly generated from the set {1, 2, …, N_p}\{i}, $(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$ or $(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G})$ is a difference vector to mutate the base vector and the $\mathbf{x}_{best,G}$ represents the best vector at the G-th generation. The parameter F is called the scale factor for scaling the difference vector and typically ranged on interval [0.4, 1.0] according to (Das & Suganthan (2011)).

Crossover: After mutation, the crossover operation is applied to each pair of a target vector $\mathbf{x}_{i,G}$ and its corresponding mutant vector $\mathbf{v}_{i,G}$ to generate a trial vector $\mathbf{u}_{i,G} = (u_{1i,G}, u_{2i,G}, \ldots, u_{Di,G})$. The widely used one is the binomial crossover operation defined as follows:

$$u_{ji,G} = \begin{cases} v_{ji,G}, & \text{if } (rand_j \leq Cr) \text{ or } (j = j_{rand}) \\ x_{ji,G}, & \text{otherwise} \end{cases} \quad (6)$$

where j = 1, 2, …, D and Cr is the crossover rate within the range $(0,1)$. In (6), $rand_j$ is a random number within the range of (0, 1) and $j_{rand} \in \{1, 2, \ldots, D\}$ is a randomly chosen index, which ensures that the trial vector $\mathbf{u}_{i,G}$ gets at least one element from $\mathbf{v}_{i,G}$.

Selection: The selection operation selects the better one from the target vector $\mathbf{x}_{i,G}$ and the trial vector $\mathbf{u}_{i,G}$, according to their fitness values is as follows:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G}, & \text{if } f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G}, & \text{otherwise} \end{cases} \quad (7)$$

Using this greedy selection scheme, all individuals of the next generation are better than the individuals of the current population.

2.2 DE with Self-adaptation Schemes

There are many ADE variants in the DE literature. This section briefly reviews jDE (Brest et. al. (2006)), JADE (Zhang & Sanderson (2009)), and MDE_pBX (Islam et. al. (2012)), since they will be compared with the proposed approach later in this study.

1) *jDE:* The scale factor and crossover rate are encoded with the individual. Brest et al. (2006) believed that better control parameter values lead to better individuals that in turn are more likely to survive. The control parameters are updated as follows:

$$F_{i,G+1} = \begin{cases} rand(0.1, 1), & \text{if } rand_1 \leq \tau_1 \\ F_{i,G}, & \text{otherwise} \end{cases} \quad (8)$$

$$Cr_{i,G+1} = \begin{cases} rand(0, 1), & \text{if } rand_2 \leq \tau_2 \\ Cr_{i,G}, & \text{otherwise} \end{cases} \quad (9)$$

where $rand(a, b)$ is a uniform random number between a and b, τ_1 = 0.1 and τ_2 = 0.1. By this way, a successful F and Cr value has the probability of 0.9 to be selected to generate an offspring at the next generation. Here a successful F and Cr value means that the offspring generated with this F and Cr value successfully enters the next generation.

2) *JADE:* Zhang and Sanderson (2009) implemented the following two mutation strategies:
"DE/current-to-$p$best/1 (without archive)":

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F(\mathbf{x}^p_{best,G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}), \quad (10)$$

"DE/current-to-$p$best/1 (with archive)":

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F(\mathbf{x}^p_{best,G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r1,G} - \tilde{\mathbf{x}}_{r2,G}), \quad (11)$$

where $\mathbf{x}^p_{best,G}$ is a $p$best solution, which is randomly chosen as one of the top 100p% individuals in the current population with $p \in (0, 1]$. Note that the parameter p determines the greediness of the mutation strategy. Denote **A** as the archive used to store the inferior solutions recently explored in the evolutionary search and **P** as the current population. Vectors $\mathbf{x}_{i,G}$, $\mathbf{x}_{r1,G}$, $\mathbf{x}_{r2,G}$, and $\mathbf{x}^p_{best,G}$ are randomly chosen from the current population **P**, but $\tilde{\mathbf{x}}_{r2,G}$ is randomly chosen from the union, $\mathbf{P} \cup \mathbf{A}$, of the current population and archive.

At each generation, the mutation factor F_i of each target vector $\mathbf{x}_i$ is independently generated according to a Cauchy distribution as;

$$F_i = Cauchy(\mu_F, 0.1), \quad (12)$$

and then truncated to be 1 if $F_i > 1$ or regenerated if $F_i \leq 0$. Denote S_F as the set of all successful mutation factors at current generation G. The location parameter μ_F is initialized to be 0.5 and then updated at the end of each generation by

$$\mu_F = c_1 \cdot \mu_F + (1 - c_1) \cdot mean_L(S_F), \quad (13)$$

where $c_1 \in [0, 1]$ controls the rate of parameter adaptation and $mean_L(\cdot)$ is the Lehmer mean given by

$$mean_L(S_F) = \sum_{F \in S_F} F^2 / \sum_{F \in S_F} F. \quad (14)$$

Analogously, the crossover rate Cr_i of each individual is independently generated according to a Gaussian distribution as;

$$Cr_i = Gaussian(\mu_{Cr}, 0.1), \quad (15)$$

and then truncated to [0, 1]. Denote S_{Cr} as the set of all successful crossover rates at current generation G.

The mean μ_{Cr} is also initialized to be 0.5 and then updated at the end of each generation by

$$\mu_{Cr} = c_2 \cdot \mu_{Cr} + (1 - c_2) \cdot mean_A(S_{Cr}), \quad (16)$$

where $c_2 \in [0,1]$ controls the rate of parameter adaptation and $mean_A(\cdot)$ is the arithmetic mean.

As JADE (Zhang & Sanderson (2009)), the Cauchy distribution is more helpful than the Normal distribution to diversify the mutation factors and thus avoid premature convergence, which often occurs in greedy mutation strategies if the mutation factors are highly concentrated around a certain value, besides, an arithmetic mean of S_F tends to be smaller than the optimal value of the mutation factor and thus it might cause premature convergence at the end. The Lehmer mean in (13) therefore is helpful to propagate larger mutation factors, which in turn improves the progress rate.

3) MDE_pBX: Unlike JADE, Islam et. al. (2012) developed the following novel mutation and crossover strategies:

"DE/current-to-gr_best/1":

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F(\mathbf{x}_{gr\_best,G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \quad (17)$$

where $\mathbf{x}_{gr\_best,G}$ is the best of the q% vectors randomly chosen from the current population and $q \in$ (0, 100), whereas $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ are two dintinct vectors and none of them is equal to $\mathbf{x}_{gr\_best,G}$ or the target vector $\mathbf{x}_{i,G}$ to ensure that none of the vectors is equal in (17). Besides, the p-best crossover operation incorporates a greedy parent selection strategy with the conventional binomial crossover scheme of DE. Parameter p is linearly decreased over generations as

$$p = ceil\left[\frac{Np}{2} \cdot \left(1 - \frac{G-1}{G_{max}}\right)\right], \quad (18)$$

where G_{max} represents the maximal number of generations and $ceil(\cdot)$ is the "ceiling" function returning the lowest integer greater than its argument.

In MDE_pBX, the control parameters, F_i and Cr_i, of each target vector are also generated according to (12) and (15), respectively. The location parameter μ_F is still initialized to be 0.5, but is updated at the end of each generation by

$$\mu_F = w_F \cdot \mu_F + (1 - w_F) \cdot mean_{power}(S_F), \quad (19)$$

where the weight term $w_F = 0.8 + 0.2 \cdot rand(0,1)$ and $mean_{power}(\cdot)$ stands for the power mean given by

$$mean_{power}(S_F) = \sum_{F \in S_F} (F^n / |S_F|)^{\frac{1}{n}}, \quad (20)$$

with $|S_F|$ denoting the cardinality of the set S_F. However, the initial value of mean μ_{Cr} becomes 0.6 and then it is updated at the end of each generation by

$$\mu_{Cr} = w_{Cr} \cdot \mu_{Cr} + (1 - w_{Cr}) \cdot mean_{power}(S_{Cr}), \quad (21)$$

where the weight term of $w_{Cr} = 0.9 + 0.1 \cdot rand(0,1)$ and the definition of power mean $mean_{power}(S_{Cr})$ is analogous to (20).

2.3 PID Controller Design

Figure 1 illustrates a standard control system with a PID controller, where $r(t)$ and $y(t)$ are the reference (desired) signal and the system output, respectively. The continuous-time form of a PID controller is described as follows:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{d}{dt} e(t), \quad (22)$$

where $e(t)$ is the error signal between the desired and actual outputs, $u(t)$ is the PID control force (output), and K_P, K_I, and K_D are PID parameters (Alfi and Modares (2011)). In the simulation, the PID control law can be discretized as follows:

$$u(k) = K_P e(k) + K_I T_s \sum_{j=1}^{k} e(j) + K_D[e(k) - e(k-1)]/T_s \quad (23)$$

where T_s is the sampling time (Zeng et. al. (2014)).

In the control system design, the objective is generally to minimize the cost function, such as the integral of absolute error $\int |e(t)| dt$ (IAE), integral of time-weighted absolute error $\int t|e(t)| dt$ (ITAE) or sum of squared error $\int e^2(t) dt$ (SSE), for measuring the control performance. Rather than using IAE, ITAE or SSE, this study selects (24) as the cost function to determine the performance of PID controller.

$$J(t) = \begin{cases} \int_0^\infty [w_1|e(t)| + w_2 u^2(t)]dt + w_3 t_r, & \text{if } \Delta y(t) \geq 0 \\ \int_0^\infty [w_1|e(t)| + w_2 u^2(t) + w_4|\Delta y(t)|]dt + w_3 t_r, & \text{if } \Delta y(t) < 0 \end{cases} \quad (24)$$

where $\Delta y(t) = y(t) - y(t - T_s)$, w_i, $i = 1, 2, 3, 4$, are weight coefficients and $w_4 \gg w_1$ (Zheng et. al. (2009); Zeng et. al. (2014)). As seen, cost function (24) could minimize the IAE. At the same time, the square of the controller output $u^2(t)$ is included to avoid exporting a large control value as well as the rise time t_r is added to hasten the transient response. Besides, in order to avoid overshooting, a penalty value is adopted in the fitness function. That is, once overshooting occurs $\Delta y(t) < 0$, the value of the overshooting (a penalty value $|\Delta y(t)|$) is added to the cost function.

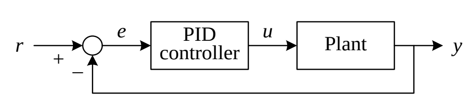


Figure 1. PID Control System.

3 ADE_$\bar{p}$BM ALGORITHM

THIS section describes the proposed ADE_$\bar{p}$BM algorithm, which includes a novel mutation strategy "DE/$\overline{p\text{best}}$/1" and a modified parameter adaptation scheme.

3.1 Mean-of-pbest Mutation Strategy

In the greedy mutation strategies, such as DE/best/k and DE/current-to-best/k, the best solution will guide the direction of the evolutionary search. The fact may lead to the problem of premature convergence caused by the reduced population diversity. "DE/current-to-$p$best/1" (Zhang & Sanderson (2009)) and "DE/current-to-gr_best/1" (Islam et. al. (2012)) are two simple but effective mutation strategies to solve the above problem. Originated from these two methodologies, this study develops another kind of less greedy and more explorative mutation strategy. The new mutation strategy is to replace the best solution in "DE/best/1" with the mean of p top-ranked vectors ($\overline{p\text{best}}$). Such a mutation strategy, termed "DE/$\overline{p\text{best}}$/1", is proposed to serve as the basis of the ADE in this study.

Denote, $\mathbf{x}^{p}_{best(k),G}$, $k = 1,2,\ldots,p$, as the kth best vector at the G-th generation. Under this denotation, $\mathbf{x}^{p}_{best(k-1),G}$ is better than $\mathbf{x}^{p}_{best(k),G}$, i.e., $f(\mathbf{x}^{p}_{best(k-1),G}) \leq f(\mathbf{x}^{p}_{best(k),G})$, where $f(\cdot)$ is the fitness function. It is also obvious that $\mathbf{x}^{p}_{best(1),G} = \mathbf{x}_{best,G}$, while $k = 1$. Once all the p-top best vectors are determined, the mean of those vectors can be represented by

$$\overline{\mathbf{x}}^{p}_{best,G} = \frac{1}{p}\sum_{k=1}^{p}\mathbf{x}^{p}_{best(k),G}. \quad (25)$$

As seen, the mean-of-$p$best vector $\overline{\mathbf{x}}^{p}_{best,G}$ involves not only the best solution information but also the information of other top-ranked solutions.

The proposed mutation strategy is a generalization of "DE/best/1", where $\overline{\mathbf{x}}^{p}_{best,G}$ plays the role of the single best solution in DE/best/1 as follows: "DE/$\overline{p\text{best}}$/1":

$$\mathbf{v}_{i,G} = \overline{\mathbf{x}}^{p}_{best,G} + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}), \quad (26)$$

with the help of the above strategy, the target solutions are not always attracted towards the single best vector, and this feature is helpful in avoiding a premature convergence at a local optima. In order to guarantee the proposed DE/$\overline{p\text{best}}$/1 and not to attract towards the single best vector, the limitation of (26) is $N_p \geq p \geq 2$. Note that the parameter p also determines the greediness of the mutation strategy as in JADE and MDE_pBX.

3.2 DE with Self-adaptation Schemes

Analogous to JADE ((Zhang & Sanderson (2009)) and MDE_pBX (Islam et. al. (2012)), the adaptation of control parameters used in this study is also based on the following principle: "Better control parameters tend to generate individuals that are more likely to survive and thus these values should be propagated to the following generations". The basic approach to implement this principle is to record recent successful scale factors and crossover rates and then use them to guide the generation of new F_i 's and Cr_i 's, respectively. The main difference between them is that the scale parameter γ_F of Cauchy distribution and the standard deviation σ_{Cr} of the Gaussian distribution are adaptable in the proposed schemes but γ_F and σ_{Cr} are constant in JADE and MDE_pBX.

Crossover Rate Adaptation: At every generation, the crossover rate Cr_i of each individual is independently generated according to a Gaussian distribution as;

$$Cr_i = Gaussian(\mu_{Cr}, \sigma_{Cr}), \quad (27)$$

and then truncated to [0, 1]. The mean μ_{Cr} is initialized to be 0.5 and then updated at the end of each generation by

$$\mu_{Cr} = \widehat{w}_{Cr} \cdot \mu_{Cr} + (1 - \widehat{w}_{Cr}) \cdot mean_A(S_{Cr}), \quad (28)$$

where $mean_A(\cdot)$ is the arithmetic mean and the weight term $\widehat{w}_{Cr}$ is randomly generated by

$$\widehat{w}_{Cr} = 1 - 0.5 \cdot rand(0,1). \quad (29)$$

While the standard deviation of the members in S_{Cr} plays the role of the mean value in (28), the adaptation of σ_{Cr} can be developed in a similar way. In this study, the standard deviation σ_{Cr} is initialized to be 0.1 and then updated at the end of each generation as

$$\sigma_{Cr} = \widehat{w}_{Cr} \cdot \sigma_{Cr} + (1 - \widehat{w}_{Cr}) \cdot std(S_{Cr}), \quad (30)$$

where $std(\cdot)$ is the standard deviation. Note that σ_{Cr} is truncated to be 0.1 if $\sigma_{Cr} > 0.1$, i.e., $\sigma_{Cr} \in (0, 0.1]$. If S_{Cr} is empty at a specific generation, μ_{Cr} remains changeless at that generation but σ_{Cr} is reset to be 0.1.

Scale Factor Adaptation: At each generation, the scale factor F_i of *each* target vector $\mathbf{x}_i$ is independently generated according to a Cauchy distribution as;

$$F_i = Cauchy(\mu_F, \gamma_F), \quad (31)$$

and then truncated to be 1 if $F_i > 1$ or regenerated if $F_i \leq 0$. That is $F_i \in (0, 1]$. The location parameter μ_F is initialized to be 0.5 and then updated at the end of each generation by

$$\mu_F = \widehat{w}_F \cdot \mu_F + (1 - \widehat{w}_F) \cdot mean_L(S_F), \quad (32)$$

where $mean_L(\cdot)$ is the Lehmer mean and

$$\widehat{w}_F = 1 - 0.5 \cdot rand(0,1). \quad (33)$$

In the Cauchy distribution, the scale parameter specifies the half-width at half-maximum, as the standard deviation in the Gaussian distribution. Thus the proposed adaptation of γ_F is similar to (30). That is, the scale parameter γ_F is initialized to be 0.1 and then updated at the end of each generation as;

$$\gamma_F = \widehat{w}_F \cdot \gamma_F + (1 - \widehat{w}_F) \cdot std(S_F), \qquad (34)$$

The scale parameter is truncated to be 0.1 if $\gamma_F > 0.1$. In another words, $\gamma_F \in (0, 0.1]$. Similarly, if the set S_F is empty at a specific generation, the location para-meter μ_F remains changeless at that generation but the scale parameter γ_F is reseted to be 0.1.

Explanations: The proposed adaptation of the control parameters is also based on the adaptation principle *used* in JADE and MDE_pBX. As seen in (28) and (32) they are analogous to (16) and (13), respectively. Besides, this study also uses the same idea to implement the adaptations of the standard deviation σ_{Cr} (30) and scale parameter γ_F (34). However, the different adaptive DE algorithm has different settings for the weight terms. JADE utilizes the constant weight term, i.e., $c = 0.9$ in (13) and (16) (Zhang and Sanderson (2009)). The experimental results in MDE_pBX indicates that small random perturbations to the weight term w in (19) and (21) are very effective in improving the search performance on a wide variety of functions (Islam et. al. (2012)). Howerer, the magnitude of random perturbation, i.e., 0.2 for the scalar factor F and 0.1 for the crossover rate Cr, must be properly specified in advance. In order to reduce the effect of the setting parameter caused by the user, the proposed weight term is simply set to be in the form of $\widehat{w} = 1 - 0.5 \cdot rand(0,1)$. It is almost parameter free. Since $0.5 \cdot rand(0,1) < 0.5$, we have $\widehat{w} > (1 - \widehat{w})$. Such a relationship is consistent with that used in JADE and MDE_pBX. For example; $c_1 > 1 - c_1$ if $c_1 = 0.9$ in (13) and $w_{Cr} > 1 - w_{Cr}$ in (21). Our experiment results also reveal that the proposed weight terms not only are insensitive to different problems according to their role of controlling the rate of the parameter adaptation but also enable ADE_$\bar{p}$BM to perform better than JADE and MDE_pBX on a wide variety of functions and PID controller design problems.

3.3 DE with Self-adaptation Schemes

Table 1 represents the procedure used for implementing the proposed ADE_$\bar{p}$BM algorithm. All the parameters of the Cauchy and Gaussian distribution functions are updated at the end of each generation according to the record of recent successful control parameters. Analogously to JADE and MDE_pBX, the basic approach to implement the adaptation of the control parameters is to record recent successful scale factors and crossover rates, then use them to guide the generation of new F_i's and Cr_i's, respectively. However, rather than using $\gamma_F = 0.1$ and $\sigma_{Cr} = 0.1$ in the probability distribution functions, both scale parameter γ_F and standard deviation σ_{Cr} are adaptable in the proposed ADE_$\bar{p}$BM algorithm. The adaptations of γ_F and σ_{Cr} are helpful to improve the solution accuracy and reliability of the ADE algorithm, shown later in the next section.

Table 1. **Pseudo Code of the ADE_$\bar{p}$BM Algorithm.**

Line Procedure of the ADE_$\bar{p}$BM
Begin
Set $\mu_F = 0.5$, $\gamma_F = 0.1$, $\mu_{Cr} = 0.5$, $\sigma_{Cr} = 0.1$
Initialize a random population $\{\mathbf{x}_{i,0} \mid i = 1,2,\ldots,Np\}$
For $G = 1$ to G_{max}
$S_F = \emptyset$, $S_{Cr} = \emptyset$
Find the best vector $\mathbf{x}_{best,G}$
Determine the mean of p-top best vectors $\bar{\mathbf{x}}^p_{best,G} = \sum_{k=1}^{p} \mathbf{x}^p_{best(k),G}/p$
For $i = 1$ to Np
Generate $F_i = Cauchy(\mu_F, \gamma_F)$, $Cr_i = Gaussian(\mu_{Cr}, \sigma_{Cr})$
Randomly generate two integers $r1$ and $r2$ in the range $[1, Np]$, and $r1 \neq r2 \neq i$
$\mathbf{v}_{i,G} = \bar{\mathbf{x}}^p_{best,G} + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$
Randomly generate a integer j_{rand} in the range $[1, D]$
For $j = 1$ to D
If $j = j_{rand}$ or $rand(0,1) < Cr_i$
$u_{ji,G} = v_{ji,G}$
Else
$u_{ji,G} = x_{ji,G}$
End If
End for
If $f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G})$
$\mathbf{x}_{i,G+1} = \mathbf{u}_{i,G}$; $F_i \rightarrow S_F$, $Cr_i \rightarrow S_{Cr}$
Else
$\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G}$
End If
End for
If $S_F \neq \emptyset$ and $S_{Cr} \neq \emptyset$
$\mu_F = \widehat{w}_F \cdot \mu_F + (1 - \widehat{w}_F) \cdot mean_L(S_F)$, $\gamma_F = \widehat{w}_F \cdot \gamma_F + (1 - \widehat{w}_F) \cdot std(S_F)$
$\mu_{Cr} = \widehat{w}_{Cr} \cdot \mu_{Cr} + (1 - \widehat{w}_{Cr}) \cdot mean_A(S_{Cr})$, $\sigma_{Cr} = \widehat{w}_{Cr} \cdot \sigma_{Cr} + (1 - \widehat{w}_{Cr}) \cdot std(S_{Cr})$
Else
$\gamma_F = 0.1$, $\sigma_{Cr} = 0.1$
End If
End for
End

Table 2. Comparison of Weight Terms and Initial Value Settings.

	JADE	MDE_pBX	ADE_$\bar{p}$BM
Weight terms:			
μ_F	$c_1 = 0.9$	$w_F = 0.8 + 0.2 \cdot rand(0,1)$	$\hat{w}_F = 1 - 0.5 \cdot rand(0,1)$
γ_F	–	–	the same value as in μ_F
μ_{Cr}	$c_2 = 0.9$	$w_{Cr} = 0.9 + 0.1 \cdot rand(0,1)$	$\hat{w}_{Cr} = 1 - 0.5 \cdot rand(0,1)$
σ_{Cr}	–	–	the same value as in μ_{Cr}
Initial value settings:			
μ_F	0.5	0.5	0.5
γ_F	**0.1**	**0.1**	0.1
μ_{Cr}	0.5	0.6	0.5
σ_{Cr}	**0.1**	**0.1**	0.1

Note: **boldface** indicates the number being a constant

4 SIMULATION RESULTS

IN this Section, the ADE_$\bar{p}$BM is applied to optimize a set of 12 benchmark functions selected from (Brest et. al. (2006)) and design the optimal PID controller for two single-variable plants. The performance of the proposed ADE_$\bar{p}$BM algorithm is compared with four state-of-the-art ADE variants: jDE (Brest et. al. (2006)), SaDE ((Qin et. al. (2009)), JADE (Zhang & Sanderson (2009)), and MDE-pBX (Islam et. al. (2012)). Besides, in the simulations, all the programs coded by Matlab version R2010a were executed by a personal computer with Intel(R) Core(TM) i5-3470 CPU @ 3.20/3.60 GHz processor, 8.0-GB RAM and Windows 7 operating system with service pack 1.

4.1 Global Optimization Problems

Table 2 represents the selected benchmark functions and detailed descriptions about those benchmark functions can be found in (Brest et. al. (2006)).

Dimensions (D), search spaces, global optimum values (f_{min}), and the maximum number of generations (G_{max}) for each test function are also listed in the same table. Population size N_p for the ADE variants has been kept to 100 irrespective of problem dimension D. The settings of weight terms and initial values of all adjustable parameters used in JADE, MDE_pBX and ADE_$\bar{p}$BM are given in Table 3. Other specific parameters of state-of-the-art ADE variants and ADE_$\bar{p}$BM are listed as follows:

1) jDE with $F_l = 0.1$, $F_u = 0.9$, and $\tau_1 = \tau_2 = 0.1$ (Brest et. al. (2006)).
2) JADE with $c = 0.9$, $p = 0.05$, and optional external archive (Zhang and Sanderson (2009)).
3) MDE_pBX with $q = 15\%$ and $n = 1.5$ Islam et. al. (2012)).
4) ADE_$\bar{p}$BM with $p = 5$.

Each algorithm was run as 50 independent trials on every benchmark function and their results are used in the comparison. This study also selects a threshold value for each benchmark function to compare the convergence speeds and reliabilities between different ADE algorithms, except function f_7, for functions with minimum at zero, this threshold is at 10^{-5}. The threshold of f_7 is 10^{-3}. For function f_8, this value is chosen to be $-12{,}000$.

In order to demonstrate the benefit of the proposed adaptations of γ_F and σ_{Cr}, the results obtained by ADE_$\bar{p}$BM with $\gamma_F = 0.1$ and $\sigma_{Cr} = 0.1$ are given in

Table 3. Twelve Selected Benchmark Functions.

Function	D	Search range	f_{min}	G_{max}
$f_1(\mathbf{x}) = \sum_{i=i}^{D} x_i^2$	30	$[-100, 100]^D$	0	1,500
$f_2(\mathbf{x}) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$	30	$[-10, 10]^D$	0	2,000
$f_3(\mathbf{x}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]^D$	0	5,000
$f_4(\mathbf{x}) = \max\{\lvert x_i\rvert, 1 \le i \le D\}$	30	$[-100, 100]^D$	0	5,000
$f_5(\mathbf{x}) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^D$	0	20,000
$f_6(\mathbf{x}) = \sum_{i=1}^{D}(\lfloor x_i + 0.5\rfloor)^2$	30	$[-100, 100]^D$	0	1,500
$f_7(\mathbf{x}) = \sum_{i=1}^{D} i x_i^4 + random[0,1)$	30	$[-1.28, 1.28]^D$	0	3,000
$f_8(\mathbf{x}) = -\sum_i^D\left(x_i \sin\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]^D$	-12569.5	9,000
$f_9(\mathbf{x}) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^D$	0	5,000
$f_{10}(\mathbf{x}) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]^D$	0	1,500
$f_{11}(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^D$	0	2,000
$f_{12}(\mathbf{x}) = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i \le a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	30	$[-50, 50]^D$	0	1,500

Table 4. Comparisons of Solution Accuracy and Success Rate for each ADE Variant.

Function		jDE	SaDE	JADE	MDE_pBx	ADE_$\bar{p}$BM	ADE_$\bar{p}$BM
		–	–	$\gamma_F = 0.1$, $\sigma_{Cr} = 0.1$	$\gamma_F = 0.1$, $\sigma_{Cr} = 0.1$	γ_F and σ_{Cr}: adaptable	$\gamma_F = 0.1$, $\sigma_{Cr} = 0.1$
f_1	Mean	1.0293e-15(+)	5.4254e-38(+)	1.7826e-64(+)	2.9293e-22(+)	**2.3496e-97**	3.9786e-119
	Std. Dev.	4.55e-16	5.37e-38	3.27e-64	2.06e-22	6.94e-97	1.28e-118
	Rate (%)	**100**	**100**	**100**	**100**	**100**	100
f_2	Mean	1.2974e-13(+)	9.9708e-27(+)	1.8412e-47(+)	1.8566e-15(+)	**2.0556e-66**	1.7184e-51
	Std. Dev.	3.79e-14	6.41e-27	1.82e-47	1.05e-15	1.04e-65	5.46e-51
	Rate (%)	**100**	**100**	**100**	**100**	**100**	100
f_3	Mean	–(+)	6.9257e-36(+)	**6.0877e-53**(+)	2.9038e-28(+)	1.4531e-30	9.7286-69
	Std. Dev.	–	1.194e-35	1.08e-52	5.15e-28	6.15e-30	2.98e-68
	Rate (%)	0	**100**	**100**	**100**	**100**	100
f_4	Mean	1.0494e-08(+)	**2.8600e-24**(+)	1.8922e-12(+)	6.2056e-16(+)	5.5128e-09	1.8568e-11
	Std. Dev.	3.71e-09	1.94e-24	3.03e-12	8.21e-16	2.66e-08	3.90e-11
	Rate (%)	**100**	**100**	**100**	**100**	**100**	100
f_5	Mean	**0(–)**	6.5263e-30(–)	–(+)	**0(–)**	2.9997e-27	1.2402e-26
	Std. Dev.	0	8.566e-30	–	0	5.73e-27	3.75e-26
	Rate (%)	**100**	**100**	2	**100**	**100**	90
f_6	Mean	**0(=)**	**0(=)**	**0(=)**	**0(=)**	**0**	0
	Std. Dev.	0	0	0	0	0	0
	Rate (%)	**100**	**100**	**100**	**100**	**100**	42
f_7	Mean	–(+)	8.3274e-04(+)	7.1478e-04(+)	7.8805e-04(+)	**6.1142e-04**	6.6889e-04
	Std. Dev.	–	1.40e-04	2.26e-04	1.59e-04	1.39e-04	1.79e-04
	Rate (%)	0	48	26	38	**84**	56
f_8	Mean	**-1.2569e+04(–)**	**-1.2569e+04(–)**	-1.2538e+04(=)	–(+)	-1.2493e+04	-1.2384e+04
	Std. Dev.	7.34e-12	7.34e-12	5.24e+01	–	1.14e+02	1.15e+02
	Rate (%)	**100**	**100**	**100**	2	**100**	100
f_9	Mean	**0(–)**	**0(–)**	**0(–)**	**0(–)**	**0**	4.8009e-17
	Std. Dev.	0	0	0	0	0	2.92e-16
	Rate (%)	**100**	**100**	**100**	96	78	74
f_{10}	Mean	9.0786e-09(+)	**4.4408e-15(=)**	6.2172e-15(+)	6.9841e-12(+)	**4.4408e-15**	8.9557e-15
	Std. Dev.	1.88e-09	0	1.81e-15	2.83e-12	0	4.67e-15
	Rate (%)	**100**	**100**	48	**100**	**100**	96
f_{11}	Mean	**0(=)**	**0(=)**	**0(=)**	**0(+)**	**0**	5.8432e-18
	Std. Dev.	0	0	0	0	0	2.51e-17
	Rate (%)	**100**	**100**	**100**	92	**100**	76
f_{12}	Mean	8.0439e-17(+)	**1.5705e-32(=)**	**1.5705e-32(=)**	1.9951e-23(+)	**1.5705e-32**	2.4036-32
	Std. Dev.	4.10e-17	5.52e-48	5.52e-48	1.89e-23	5.52e-48	2.27e-32
	Rate (%)	**100**	**100**	**100**	**100**	**100**	88
No. of best		5	7	5	4	8	–
No. of 100%		10	11	9	8	10	–
w/t/l		7/2/3	5/4/3	7/4/1	9/1/2	–	–

"+", "–" and "=" indicate ADE_$\bar{p}$BM is respectively better than, worse than, or similar to its competitor according to Wilcoxon's rank sum test at $\alpha = 0.05$.

the last column of Tables 4 and 5. The shaded numbers in the last column represent that ADE_$\bar{p}$BM with $\gamma_F = 0.1$ and $\sigma_{Cr} = 0.1$, which performs better than the ADE_$\bar{p}$BM with adaptable γ_F and σ_{Cr} on the corresponding item. Note that, unless otherwise stated, the following comparisons do not take those numerical results into consideration.

4.1.1 Comparisons on Solution Accuracy and Reliability

For each function in every ADE variant, the mean and standard deviation of all successful runs are presented in Table 4. The success rate of each algorithm is also given in the table and it is useful to compare the reliability of the different algorithms. The success of an algorithm means that this algorithm can result in a function value no worse than the predefined threshold, with the number of generations less than the corresponding maximum number. The success rate is calculated as the number of successful runs divided by the total number of runs. For clarity, **boldface** indicates the best result(s) among all of the ADE variants, and rows. "No. of best" and "No. of 100%" represent the numbers of the best mean values and 100% successful runs, respectively, the corresponding algorithm can attain. The proposed ADE_$\bar{p}$BM obtains the best accuracy on 8 out of 12 functions and yields 100% success rate on 10 functions, while SaDE performs the best accuracy on 7 functions and

Table 5. Comparisons of Convergence Speed for each ADE Variant.

Function		jDE	SaDE	JADE	MDE_pBx	ADE_$\bar{p}$BM	ADE_$\bar{p}$BM
		–	–	$\gamma_F = 0.1$, $\sigma_{Cr} = 0.1$	$\gamma_F = 0.1$, $\sigma_{Cr} = 0.1$	γ_F and σ_{Cr}: adaptable	$\gamma_F = 0.1$, $\sigma_{Cr} = 0.1$
f_1	Mean Gens	742.02	328.26	206.46	523.56	137.88	112.34
	Time (sec)	0.2171	3.8153	0.0757	0.2160	**0.0627**	0.0376
f_2	Mean Gens	951.08	474.44	291.50	822.04	200.04	174.78
	Time (sec)	0.2864	5.6459	0.1100	0.3461	**0.0920**	0.0591
f_3	Mean Gens	–	1129.18	779.32	1359.10	743.68	784.08
	Time (sec)	–	14.7294	**0.3134**	0.6604	0.3570	0.2945
f_4	Mean Gens	3483.54	1253.92	2220.34	1670.60	1212.62	2233.52
	Time (sec)	1.0207	16.3032	0.7967	0.6867	**0.5186**	0.7177
f_5	Mean Gens	6719.04	2330.16	–	2065.38	1547.10	1096.71
	Time (sec)	2.0953	33.1901	–	0.8545	**0.5367**	0.3424
f_6	Mean Gens	354.40	154.36	126.66	251.44	63.50	106.04
	Time (sec)	0.1106	1.9295	0.0461	0.1035	**0.0221**	0.0347
f_7	Mean Gens	–	2374.12	2232.38	2165.05	1652.47	1504.50
	Time (sec)	–	28.2102	1.5039	1.5744	**1.1055**	0.9465
f_8	Mean Gens	1618.54	510.48	760.40	–	1226.94	925.24
	Time (sec)	0.6387	6.9842	**0.3386**	–	0.5180	0.3630
f_9	Mean Gens	3513.54	1198.70	1479.22	2537.97	1752.38	1919.18
	Time (sec)	1.2138	15.5204	**0.5591**	1.1518	0.6692	0.6599
f_{10}	Mean Gens	1038.84	459.54	324.04	761.70	201.60	157.27
	Time (sec)	0.3573	5.7434	0.1211	0.3465	**0.0785**	0.0508
f_{11}	Mean Gens	794.82	356.92	358.16	529.13	156.22	119.13
	Time (sec)	0.3239	4.9679	0.1652	0.2698	**0.0687**	0.0470
f_{12}	Mean Gens	657.32	269.52	188.68	455.32	124.58	145.93
	Time (sec)	0.2895	3.7472	0.0889	0.2458	**0.0599**	0.0614
No. of fastest		–	–	3	–	9	–

achieves 100% success rate on 11 functions. As seen, ADE_$\bar{p}$BM and SaDE are the best two algorithms. In order to compare the significance between the two algorithms, the Wilcoxon's rank sum test is also used (Yu et. al. (2014)). In the last row of Table 4, according to the Wilcoxon's test, the results are summarized as "*w*/*t*/*l*," which denotes that ADE_$\bar{p}$BM wins *w* functions, ties in *t* function, and loses in *l* functions, compared with its competitors. For example, with SaDE, ADE_$\bar{p}$BM wins in 5 functions, ties in 4 functions, and loses 3 functions according to the Wilcoxon's test at $\alpha = 0.05$. The fact reveals that the proposed ADE_$\bar{p}$BM performs slightly better than SaDE.

Table 4 also shows the benefit of the proposed adaptations of γ_F and σ_{Cr} on solution accuracy and reliability. As far as the solution accuracy is considered, ADE_$\bar{p}$BM with $\gamma_F = 0.1$ and $\sigma_{Cr} = 0.1$ outperforms ADE_$\bar{p}$BM with adaptable γ_F and σ_{Cr} only on three functions f_1, f_3, and f_4. However, the former performs significantly worse than the latter on the reliability (success rate). The fact could reveal that the adaptations of γ_F and σ_{Cr} actually enhance the solution accuracy and reliability of the proposed ADE_$\bar{p}$BM algorithm.

4.1.2 Comparisons on Convergence Speed

Figure 2 depicts the convergence graphs for the median run of the ADE algorithms on 12 functions. As evident from the convergence characteristics, the overall convergence speed of ADE_$\bar{p}$BM seems to be the best among the contestant ADE algorithms. However, compared to JADE and MDE_pBX, ADE_$\bar{p}$BM requires two extra operations to perform the adaptations of γ_F and σ_{Cr}, i.e., (30) and (34), that cost computational time. The evidence that resulted from Figure 2 therefore cannot imply that the ADE_$\bar{p}$BM actually uses a lesser computational time.

In order to truly demonstrate the convergence speed of the algorithms, Table 5 lists the average number of generations and the corresponding computational time of each ADE variant required to reach the respective predefined threshold. **Boldface** indicates the fastest convergence speed among all of the ADE variants. Row "No. of fastest" represents the numbers of the fastest convergence speed the corresponding algorithm can, others on 9 out of 12 functions, but slower than JADE only on the remaining three problems.

To sum up, the proposed ADE_$\bar{p}$BM performs the best accuracy on 8 functions, yields 100% success rate on 10 functions, and attains the fastest convergence speed on 9 functions. That is to say, ADE_$\bar{p}$BM can perform better accuracy, reliability and efficiency than the other ADE variants on a wide variety of functions.

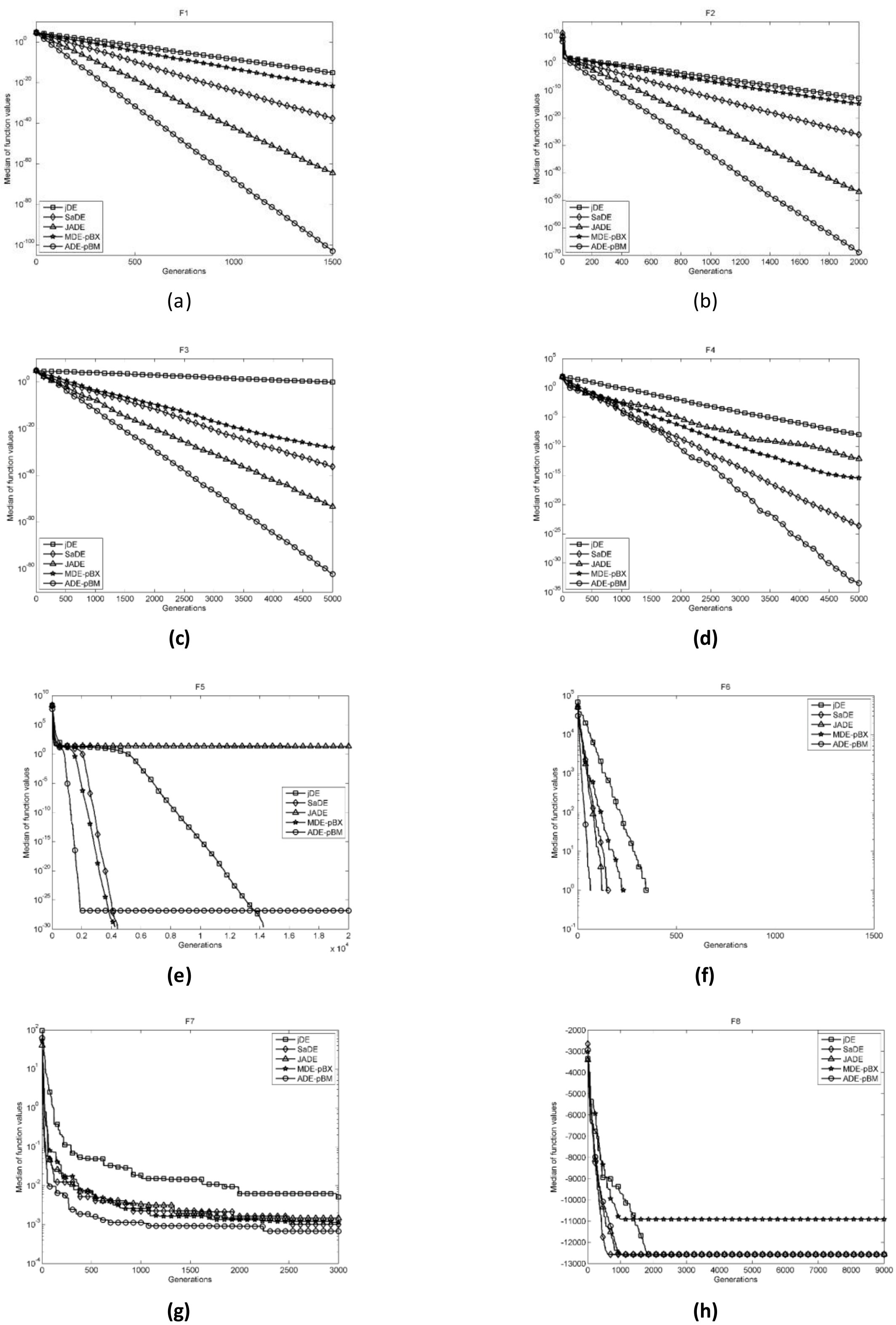
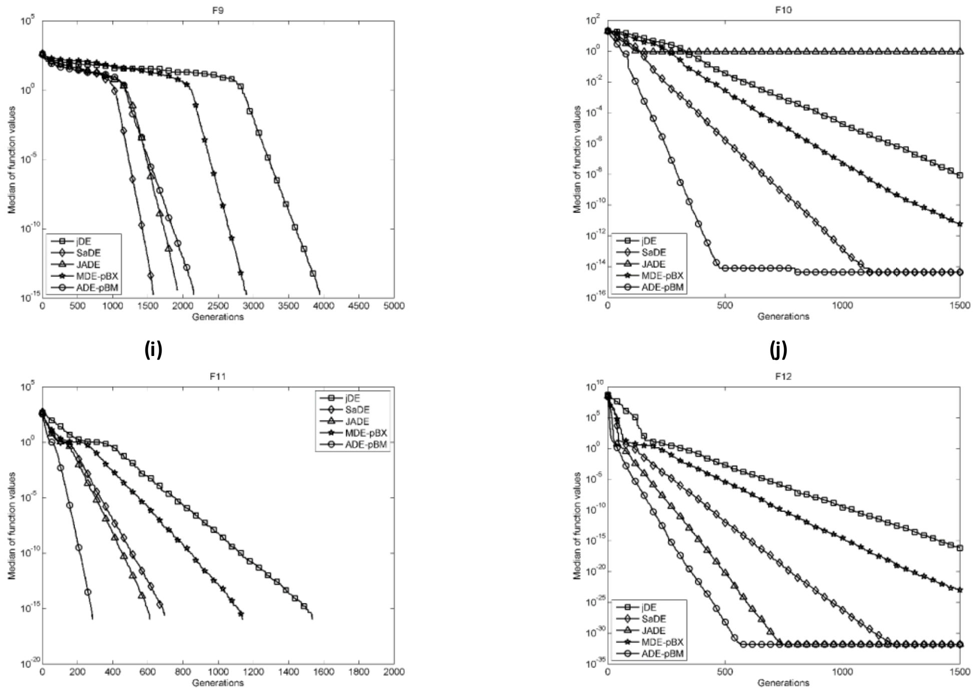


Figure 2. PID control system. The Convergence Curve of Median Run of Five Algorithms over 12 Test Functions. (a) f_1. (b) f_2. (c) f_3. (d) f_4.



(i) (j)

(k) (l)

(*Continued.*) **Figure 2. PID control system. Convergence Curve of Median Run of Five Algorithms over 12 Test Functions. (e) f_5. (f) f_6. (g) f_7. (h) f_7. (i) f_9. (j) f_{10}. (k) f_{11}. (l) f_{12}.**

4.2 PID Controller Designs

Two single-variable plants are chosen to demonstrate the search performance of the proposed ADE_$\bar{p}$BM algorithm in the tuning of the PID parameters. The transfer functions of the chosen plants are given as follows:

$$\text{Plant 1: } G_1(s) = \frac{1.6}{s^2 + 2.584s + 1.6}, \tag{35}$$

$$\text{Plant 2: } G_2(s) = \frac{15}{50s^3 + 43s^2 + 3s + 1}, \tag{36}$$

With the system sampling time being 0.05 second and the range of the control value u being $[-10, 10]$, other relevant system variables are; $K_P \in [0, 20]$, $K_I \in [0, 20]$, and $K_D \in [0, 10]$. In order to compare with the simulation results obtained in (Zheng et. al. (2009); Zeng et. al. (2014)), this study adopts the same fitness function, i.e., the cost function given in (24), and the corresponding weight coefficients are also set as; $w_1 = 0.999$, $w_2 = 0.001$, $w_3 = 2.0$, and $w_4 = 50$.

In the simulations, the step response of each PID control system tuned by ADE_$\bar{p}$BM is compared with that tuned by the previous four state-of-the-art ADE variants as well as the self-organizing genetic algorithm (SOGA) (Zheng et. al. (2009)), adaptive genetic algorithm (AGA) (Zhang et. al. (2007)), probability-based binary particle swarm optimization (PBPSO) (Menhas et. al. (2014)) and binary-coded extremal optimization algorithm (BCEO) (Zeng et. al. (2014)). For each algorithm, the population size is 50 and the maximum number of generations is 100. Except for these two parameters, other relevant parameters of ADE_$\bar{p}$BM and four contestants ADE variants are the same as previous settings. The following parameters are used for SOGA; the size of the dominant population is 10, the crossover rate P_c is 0.6, and the mutation rate function is defined as:

$$P_m(t) = \frac{\alpha\,[t - (k + 0.5)T_c]^2}{T_c^2}, \tag{37}$$

where t is the current generation number, $T_c = 50$ indicates the mutation period, $\alpha = 4$ represents a mutation turning coefficient, and k is the number of the period. The parameters of AGA are $P_c = 0.7$ and $P_m = 0.01$. Each algorithm ran for 20 independent trials on every transfer function and the median run of each algorithm is used in the comparison. The performance of these algorithms are evaluated by the indices including the best fitness (J_{best}), maximal overshoot (PO%), rising time (T_{rise}), steady-state error (e_{ss}%), settling time with 0.1% error ($T_{set}^{0.1}$) and running time (T_{run}).

The optimal parameters of PID controllers and the experimental results obtained by different algorithms for plant 1 are shown in Table 6. Note that **boldface** and *italic* (if have) in the table indicate the best and near-best results, respectively. In addition, the optimal PID parameters attained by SOGA, AGA, PBPSO and BCEO are previous findings given by Zeng et. al.

(2014)). The corresponding running times therefore are not shown in the table. Figure 3 depicts step responses for plant 1 under different algorithms-based PID controllers. Since the optimal parameters of the PID controllers and the corresponding best fitness values obtained by different ADE variants are almost the same, their step response curves are close to each other. Therefore Figure 3 only illustrates the step response obtained by the proposed ADE_$\bar{p}$BM. It can be seen from Figure 3 and Table 6 that the PID controller tuned by ADE_$\bar{p}$BM has the minimum overshoot, the smallest settling time and the least running time as well as the near-best fitness and a small steady-state error of 1.4×10^{-5}%. The results indicate that ADE_$\bar{p}$BM can attain better transient and steady-state performances.

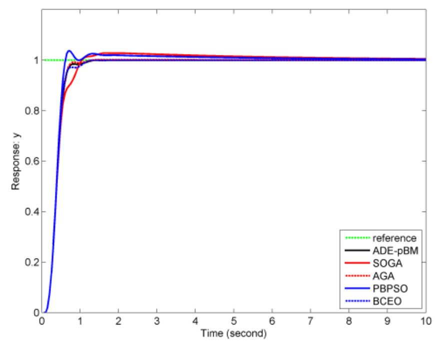


Figure 3. Step Response for Plant 1 with Different Algorithm-based PID Controllers.

The parameters of the PID controllers and the experimental results for plant 2 are given in Table 7, while the corresponding step responses are depicted in Figure 4. Although T_{rise} and e_{ss}% obtained by MDE_pBX are all smaller than those by other algorithms, MDE_pBX performs a large PO% and a big J_{best}. Compared with the others, it can be observed that the PID controllers tuned by ADE_$\bar{p}$BM and SaDE show almost the same performance indices including J_{best}, PO%, T_{rise}, e_{ss}%, $T_{set}^{0.1}$, and T_{run}. Overall speaking, they are the best two algorithms but ADE_$\bar{p}$BM is slightly better than SaDE.

5 CONCLUSIONS

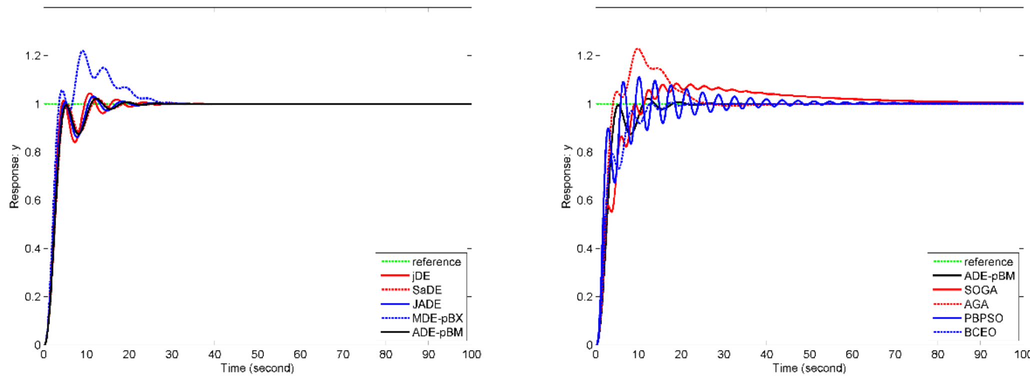
BY replacing the best solution in "DE/best/1" with the mean of p top-ranked vectors, this study proposed a less greedy and more explorative mutation strategy "DE/$\bar{p}$best/1" to avoid premature convergence at local optima. Besides, the parameter adaptation of ADE_$\bar{p}$BM was implemented by evolving the scale factor F and crossover rate Cr based on their historical record of success. Four parameters, σ_{Cr}, μ_{Cr}, γ_F, and μ_F, are updated at the end of each generation according to the record of recent successful control parameters. The purpose of this parameter adaptation mechanism is to improve the solution accuracy and robustness of the algorithm. The results of global optimization problems show that the ADE_$\bar{p}$BM could perform better accuracy, reliability and efficiency than the four state-of-the-art adaptive DE variants on a wide variety of functions.

Table 6. Optimal Parameters of the PID Controllers and Performance Indices: Plant 1

Algorithm	K_P	K_I	K_D	J_{best}	PO%	T_{rise}	e_{ss}%	$T_{set}^{0.1}$	T_{run}
jDE	15.5558	2.6382	3.5242	9.6329	3.14e-3	*0.65*	3.14e-3	**1.35**	*33.5734*
SaDE	15.6506	2.6405	3.5522	*9.6074*	3.22e-4	*0.65*	2.89e-5	**1.35**	34.0621
JADE	15.6657	2.6394	3.5539	9.6142	8.14e-4	*0.65*	8.14e-4	**1.35**	34.0727
MDE_pBX	15.6574	2.6404	3.5537	**9.6073**	*2.15e-4*	*0.65*	**1.08e-8**	**1.35**	34.1400
ADE_$\bar{p}$BM	15.6119	2.6412	3.5428	*9.6092*	**1.74e-4**	*0.65*	1.43e-5	**1.35**	**33.5433**
SOGA	19.390	4.119	5.151	14.2753	2.81e+0	0.90	5.14e-1	>10	–
AGA	15.2884	2.7566	3.4506	9.8644	2.12e-1	*0.65*	4.70e-2	5.75	–
PBPSO	19.9990	42174	3.9562	22.2145	1.67e+1	**0.60**	2.29e-1	>10	–
BCEO	17.5171	2.639	3.9296	9.7605	7.70e-3	*0.65*	1.58e-3	1.55	–

Table 7. Optimal Parameters of the PID Controllers and Performance Indices: Plant 2

Algorithm	K_P	K_I	K_D	J_{best}	PO%	T_{rise}	e_{ss}%	$T_{set}^{0.1}$	T_{run}
jDE	1.2805	0.0245	3.7686	83.2111	4.37	3.90	5.55e-3	33.1	98.1229
SaDE	0.8640	0.0220	2.8985	**80.4329**	*2.09*	4.50	3.89e-4	*28.4*	97.1462
JADE	1.0531	0.0231	3.3087	81.0380	3.05	4.20	9.88e-4	29.8	**96.5950**
MDE_pBX	1.7466	0.2183	5.3817	103.7465	22.03	**3.35**	**3.19e-6**	39.4	98.2078
ADE_$\bar{p}$BM	0.9175	0.0221	3.0456	*80.4897*	*2.17*	4.45	1.40e-4	**27.7**	*96.7331*
SOGA	2.98	0.096	12.7	156.5308	8.45	8.45	4.54e-1	>100	–
AGA	1.3294	0.1955	4.6921	108.0184	22.97	*3.75*	2.09e-5	43.15	–
PBPSO	4.0043	0.0355	10.0	165.4567	11.13	5.70	2.37e-1	>100	–
BCEO	1.7986	0.0196	6.3441	92.7663	**0.49**	7.75	1.22e-3	*29.95*	–



(a) ADE Algorithms **(b) ADE_$\bar{p}$BM, GAs, PSO and BCEO**

Figure 4. **Step Response for Plant 2 with Different Algorithm-based PID Controllers.**

On the other hand, as far as the PID controller designs on two single-variable plants are considered, the results also show that the ADE_$\bar{p}$BM can attain better transient and steady-state performance than other ADE variants.

5.1 Acknowledgment

THIS work was supported by Ministry of Science and Technology, Taiwan, R.O.C. through Grant MOST 103-2221-E-262-026 and MOST 104-2221-E-262-009.

6 REFERENCES

Abbasa, Q., Ahmadb, J., & Jabeena, H. (2018). Random controlled pool base differential evolution algorithm (RCPDE), Intelligent *Automation and Soft Computing*,24(2), 377-390.

Alfi, A., & Modares, H. (2011). System identification and control using adaptive particle swarm optimization. *Applied Mathematical Modelling*, *35*(3), 1210–1221.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Žumer, V. (2006). Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, *10*(6), 646–657.

Das, S., & Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*(1), 4–31.

Islam, Sk. M., Das, S., Ghosh, S., Roy, S., & Suganthan, P. N. (2012). An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(2), 482–500.

Kim, J. H., Lee, J., & Oh, Y. (2018). *Performance analysis for bounded persistent disturbances in PD/PID-controlled robotic systems with its experimental demonstrations: International Journal of Control: Vol 91, No 3*. *91*(3), 688–705.

Li, X., Hu, C., & Yan, X.-F. (2013). *Chaotic differential evolution algorithm based on competitive coevolution and its application to dynamic optimization of chemical processes*. *19*(1), 85–98.

Lu, X., Tang, K., Sendhoff, B., & Yao, X. (2014). A new self-adaptation scheme for differential evolution. *Neurocomputing*, *146*, 2–16.

Mallipeddi, R., Suganthan, P. N., Pan, Q., & Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, *11*(2), 1679–1696.

Menhas, M. I., Wang, L., Fei, M., & Pan, H. (2012). Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design. *Expert Systems with Applications*, *39*(4), 4390–4401.

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2008). *Differential evolution algorithm with strategy adaptation for global numerical optimization*. *13*(2), 398–417.

Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

Tabatabaei, M., & Barati-Boldaji, R. (2017). Non-overshooting PD and PID controllers design. *Automatika*, *58*(4), 400–409.

Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, *15*(1), 55–66.

Xue, Y., Zhong, S., Ma, T., & Cao, J. (2015). A hybrid evolutionary algorithm for numerical optimization problem. *Intelligent Automation & Soft Computing*, *21*(4), 473–490.

Yu, W.-J., Shen, M., Chen, W.-N., Zhan, Z.-H., Gong, Y.-J., Lin, Y., Zhang, J. (2014). Differential evolution with two-level parameter adaptation. *IEEE Transactions on Cybernetics*, *44*(7), 1080–1099.

Zeng, G.-Q., Lu, K.-D., Dai, Y.-X., Zhang, Z.-J., Chen, M.-R., Zheng, C.-W., Peng, W.-W. (2014). Binary-coded extremal optimization for the design of PID controllers. *Neurocomputing*, *138*, 180–188.

Zhang, J., Chung, H. S.-H., & Lo, W.-L. (2007). Clustering-based adaptive crossover and mutation probabilities for genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *11*(3), 326–335.

Zhang, J. H., Zhuang, J., Du, H., & Wang, S. (2009). Self-organizing genetic algorithm based tuning of PID controllers. *Information Sciences*, *179*(7), 1007–1018.

Zhang, J. Q., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, *13*(5), 945–958.

7 NOTES ON CONTRIBUTORS

T.H. Chen received a Ph.D. degree in Electrical Engineering from Tatung University, Taipei, Taiwan, R.O.C. He is currently an assistant professor with the Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan, Taiwan, R.O.C. His researches interests include evolutionary computation, fuzzy logic system, and optimal control.

Ming-Feng Yeh received his B.S., M.S., and Ph. D. degrees in Electrical Engineering from Tatung University, Taipei, Taiwan in 1993, 1995, and 1999, respectively. Since 2001, he has been with the Department of Electrical Engineering, Lunghwa University of Science and Technology, Taoyuan, Taiwan, currently a professor. His current interests include; grey system theory, neural networks, evolutionary algorithm, soft computing, and applications on bioengineering, pattern recognition, automatic control, and intelligent control.