

An Immune Self-adaptive Differential Evolution Algorithm with Application to Estimate Kinetic Parameters for Homogeneous Mercury Oxidation^{*}

HU Chunping (胡春平) and YAN Xuefeng (颜学峰)**

Automation Institute, East China University of Science and Technology, Shanghai 200237, China

Abstract A new version of differential evolution (DE) algorithm, in which immune concepts and methods are applied to determine the parameter setting, named immune self-adaptive differential evolution (ISDE), is proposed to improve the performance of the DE algorithm. During the actual operation, ISDE seeks the optimal parameters arising from the evolutionary process, which enable ISDE to alter the algorithm for different optimization problems and improve the performance of ISDE by the control parameters' self-adaptation. The performance of the proposed method is studied with the use of nine benchmark problems and compared with original DE algorithm and other well-known self-adaptive DE algorithms. The experiments conducted show that the ISDE clearly outperforms the other DE algorithms in all benchmark functions. Furthermore, ISDE is applied to develop the kinetic model for homogeneous mercury (Hg) oxidation in flue gas, and satisfactory results are obtained.

Keywords differential evolution, immune system, evolutionary computation, parameter estimation

1 INTRODUCTION

Differential evolution (DE) [1] is a new generation evolutionary algorithm (EA) and has been successfully applied to solve a wide range of optimization problems. Differential evolution is stochastic, population-based, and direct search algorithm for globally optimizing functions with real valued parameters. In the first international IEEE Competition on evolutionary optimization, DE proved to be one of the fastest EAs. Storn and Price [1] have compared DE with adaptive simulated annealing [2], the annealed Nelder and Mead approach [3], the breeder genetic algorithm [4], the EA with soft genetic operators [5], and the method of stochastic differential equations [6]. In most instances, DE outperformed all of the above minimizations approaches. Vesterstrom and Thomsen [7] applied 34 widely used benchmark problems to evaluate the performance of DE, particle swarm optimization (PSO) [8], and EA. Their study shows that DE generally outperforms the other algorithms. Krink *et al.* [9] introduced three search approaches [genetic algorithm (GA) [10], PSO, and DE] to develop bank rating systems, respectively, and turned out that DE is clearly and consistently superior compared with GA and PSO both in respect to precision and reliability.

Although previous studies have shown that DE is an efficient, effective, and robust evolutionary optimization method, DE is still in its infancy and can most probably be improved [1]. Differential evolution creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness value. Differential evolution has

three control parameters: amplification factor of the differential variation- F , crossover control parameter-CR, and population size-NP [11]. The effectiveness, efficiency, and robustness of the DE algorithm are sensitive to the settings of control parameters with various degrees [1, 12]. In the original DE, the control variables are kept fixed during the optimization. Finding the optimal value for parameters is a difficult task as the value is problem specific, especially when one wants to strike a balance between reliability and efficiency. As a solution, self-adaptation was incorporated to adjust evolutionary parameters, in which the control parameters are not required to be prespecified. With population evolution, the parameter settings are gradually self-adapted according to the learning experience. The choice of NP, which is typically regarded as only a "fine tuning parameter" is initially suggested to be $NP = 5 - 10n$ [13, 14]. Thus, self-adaptation is usually applied to F and CR control parameters.

In recent years, researchers have developed various strategies for adjusting control parameters dynamically. Abbass [15] proposed a self-adaptive Pareto DE (SPDE). In SPDE, the mutation rate is sampled for each individual from a Gaussian distribution, $N(0, 1)$. The crossover rate is first initialized for each individual from a uniform distribution, $U(0, 1)$. Then, CR is adapted as

$$CR_i(G) = CR_{r_1}(G) + N(0,1) \times [CR_{r_2}(G) - CR_{r_3}(G)] \quad (1)$$

where $r_1 \neq r_2 \neq r_3 \neq i$ and $r_1, r_2, r_3 \sim U(1, \dots, NP)$. This approach is also used in Omran *et al.* [16] to dynamically adapt the control parameters. In Ref. [16], the mutation operator changes as follows:

Received 2008-07-28, accepted 2008-12-24.

^{*} Supported by the National Natural Science Foundation of China (20506003, 20776042) and the National High-Tech Research and Development Program of China (2007AA04Z164).

^{**} To whom correspondence should be addressed. E-mail: yan_xuefeng@hotmail.com

$$\hat{\mathbf{x}}_i^{G+1} = \mathbf{x}_{r_1}^G + F_i(G) \cdot (\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G), \quad (2)$$

where

$$F_i(G) = F_{r_4}(G) + N(0, 0.5) \times [F_{r_5}(G) - F_{r_6}(G)] \quad (3)$$

in which $r_4 \neq r_5 \neq r_6$ and $r_4, r_5, r_6 \sim U(1, \dots, \text{NP})$. For the crossover rate, $\text{CR} \sim N(0.5, 1.5)$. Liu and Lampinen [12] introduced a fuzzy adaptive DE, which uses fuzzy logic controllers to adapt the search parameters for mutation operation and crossover operation. Yuan *et al.* [17] proposed a method where chaos theory was applied to obtain self-adaptive parameter settings in DE. Nobakhti and Wang [18] proposed a simple randomized self-adaptive scheme for adjusting the mutation amplification factor F to improve the DE algorithm.

In self-adaptation parameter control, the idea of an evolutionary search can be used to implement the self-adaptation of search parameters [19]. In other words, the concept of coevolution can be used to adapt the control parameters. Coevolution method is an effective approach to decompose complex structure and achieve better performance. Several applications of coevolution method, which have been proven to be useful, were described in the literatures [20–22]. In this article, the immune concepts and methods are applied to determine the parameter setting of DE. Further, the proposed method, named immune self-adaptive differential evolution (ISDE), is compared with the versions of DE proposed by Price and Storn [23], the SPDE proposed by Abbass [15], and the self-adaptive DE (SDE) proposed by Omran *et al.* [16].

For illustration, ISDE was applied to develop the kinetic model for homogeneous mercury (Hg) oxidation in flue gas. Homogeneous mercury oxidation in flue gas is a highly nonlinear reaction with reference to optimal operating conditions with many equality and inequality constraints. The kinetic model involves five reactions. Two of these reactions are reversible and three are irreversible. The preexponential factor and activation energy values in the rate constant term for each reaction need to be determined. Then, ISDE was used to determine the kinetic parameters for homogeneous mercury oxidation in flue gas with the data obtained in a laboratory scale apparatus, reported by Agarwal *et al.* [24, 25], and the kinetic model with good precision for homogeneous mercury oxidation in flue gas was developed.

2 DIFFERENTIAL EVOLUTION ALGORITHM

In order to find the global optimal solution, DE performs a group-based search instead of the point-to-point search. The group, which contains a number of solution points, is named population and represented by $\mathbf{P}(G)$ with G denoting the number of generations. The current generation $\mathbf{P}(G)$ evolves into the next generation $\mathbf{P}(G+1)$ through the operation principles. In this way, $\mathbf{P}(G)$ continually evolves along with the proceeding of G till a global or near-global optimal

solution is obtained. The solution points in the population, called individuals or chromosomes, are represented by $\mathbf{x}_1^G, \mathbf{x}_2^G, \dots, \mathbf{x}_{\text{NP}}^G$ with NP denoting the number of individuals in the population. The value of NP usually keeps unchanged in the whole evolution process. The individual $\mathbf{x}_i^G$ consists of n variables, called genes, namely: $x_{i1}, x_{i2}, \dots, x_{in}$.

Assume that the working vector containing n optimized variables is denoted by $\mathbf{x}$. The elements of the working vector $\mathbf{x}$ are designated as working parameters, denoted by $x_1, x_2, \dots, x_n$. Thus, the problem of searching for a minimum could be described as

$$\min_{\mathbf{x}} f(x_1, x_2, \dots, x_n) \quad x_i \in (x_{\text{low},i}, x_{\text{high},i}) \quad (4)$$

$i = 1, 2, \dots, n,$

where f denotes the function relating the value of optimized variables $\mathbf{x}$ and the dependent variable, which is subject to be optimized. An n -dimensional space S_0 is defined within the region $\{(x_{\text{low},i}, x_{\text{high},i}) \mid i = 1, 2, \dots, n\}$.

The procedure of executing DE can be described in the following:

(1) Initialization operation: Generate the initial individuals $\mathbf{x}_i^0$, $i = 1, 2, \dots, \text{NP}$ randomly in S_0 . Determine the mutation probability F , the crossover probability CR, and the maximal number of generations G_m . Set the current generation $G = 0$.

(2) For each individual $\mathbf{x}_i^G$, $i = 1, 2, \dots, \text{NP}$, do steps 3–5 to produce the population for the next generation $G+1$.

(3) Mutation operation: a perturbed individual $\hat{\mathbf{x}}_i^{G+1}$ is generated as follows:

$$\hat{\mathbf{x}}_i^{G+1} = \mathbf{x}_{r_1}^G + F \cdot (\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G) \quad (5)$$

with $r_1, r_2, r_3 \in [1, 2, \dots, \text{NP}]$, integer and mutually different, and $F > 0$. The integers r_1 , r_2 , and r_3 are chosen randomly from the interval $[1, \text{NP}]$ and are different from the running index i . F is a real constant scaling factor within $[0, 2]$ which controls the amplification of the differential variation $(\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G)$.

(4) Crossover operation: The perturbed individual $\hat{\mathbf{x}}_i^{G+1} = [\hat{x}_{i1}^{G+1}, \hat{x}_{i2}^{G+1}, \dots, \hat{x}_{in}^{G+1}]$ and the current individual $\mathbf{x}_i^G = [x_{i1}^G, x_{i2}^G, \dots, x_{in}^G]$ are selected by a binomial distribution to perform the crossover operation to generate the offspring $\bar{\mathbf{x}}_i^{G+1}$, namely

$$\bar{x}_{it}^{G+1} = \begin{cases} x_{it}^G, & P_t > \text{CR}, \\ \hat{x}_{it}^{G+1}, & \text{otherwise}, \end{cases} \quad t = 1, 2, \dots, n, \quad (6)$$

where P_t is a uniform random number in range $[0, 1]$, and CR is the crossover constant $\in [0, 1]$.

(5) Evaluation operation: The offspring $\bar{\mathbf{x}}_i^{G+1}$ competes one-to-one with its parent $\mathbf{x}_i^G$. The evaluation operation is expressed as

$$\mathbf{x}_i^{G+1} = \begin{cases} \bar{\mathbf{x}}_i^{G+1}, & f(\bar{\mathbf{x}}_i^{G+1}) \leq f(\mathbf{x}_i^G) \\ \mathbf{x}_i^G, & \text{otherwise.} \end{cases} \quad (7)$$

(6) $G = G + 1$.

(7) Repeat steps 2–6 as long as the number of generations is smaller than the allowable maximum number G_m and the best individual is not obtained.

The mutation strategy described above is known as DE/rand/1, meaning that the vector to be perturbed is randomly chosen, and that the perturbation consists of one weighted difference vector. DE/rand/1 is the most successful and the most widely used strategy [14]. Other useful strategies are:

“DE/best/1”: $\hat{\mathbf{x}}_i^{G+1} = \mathbf{x}_{\text{best}}^G + F \cdot (\mathbf{x}_{r_1}^G - \mathbf{x}_{r_2}^G)$.

“DE/rand-to-best/1”:

$$\hat{\mathbf{x}}_i^{G+1} = \mathbf{x}_i^G + F \cdot (\mathbf{x}_{\text{best}}^G - \mathbf{x}_i^G) + F \cdot (\mathbf{x}_{r_1}^G - \mathbf{x}_{r_2}^G).$$

“DE/best/2”:

$$\hat{\mathbf{x}}_i^{G+1} = \mathbf{x}_{\text{best}}^G + F \cdot (\mathbf{x}_{r_1}^G - \mathbf{x}_{r_2}^G) + F \cdot (\mathbf{x}_{r_3}^G - \mathbf{x}_{r_4}^G).$$

“DE/rand/2”:

$$\hat{\mathbf{x}}_i^{G+1} = \mathbf{x}_{r_1}^G + F \cdot (\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G) + F \cdot (\mathbf{x}_{r_4}^G - \mathbf{x}_{r_5}^G),$$

where $\mathbf{x}_{\text{best}}^G$ is the individual vector with best fitness value in the population at generation G . A more detailed description of each strategy is given in Ref. [13].

3 IMMUNE SELF-ADAPTIVE DIFFERENTIAL EVOLUTION

In self-adaptation parameter control, the idea of an evolutionary search can be used to implement the self-adaptation of search parameters. The parameters to be adapted are coded into the chromosomes that undergo mutation and recombination. Better values for these encoded parameters are supposed to result in better individuals that in turn are more likely to survive and produce offspring and hence propagate better parameter values [19]. In other words, self-adaptation is a strategy in which the idea of an evolutionary search was used to choose the optimal parameters.

Immune system, a highly evolved biological system with learning, memory, and pattern recognition capabilities [26], has been successfully integrated into many other evolution algorithms [27–29]. In our work, the immune concepts and methods are applied to determine the parameter setting of DE. To be exact, the aim of leading immune concepts and methods into DE is theoretically to use the previous state information of search for seeking the optimal parameters, F and CR. During the actual operation, ISDE seeks the optimal parameters arising from the evolutionary process, which enable ISDE to alter the algorithm for different optimization problems and improve the performance of ISDE by the control parameters' self-adaptation. In ISDE, the first initial antibodies are randomly generated within the feasible range. The two parameters of each individual are initialized from a normal distribution within the feasible range. The affinity values of the antibodies are calculated. Then, depending on the

affinity values, the parameters are replaced by antibody with a certain probability defined previously. In each generation, a percentage of antibodies in the antibody population are replaced by created new antibodies. Thus, the coevolution method is established. Differential evolution is used to perform evolution search in spaces of solutions, and immune system is used to perform evolution search in spaces of control parameters. The solutions and control parameters evolve interactively and selfadaptively, and both the satisfactory solutions and suitable control parameters can be obtained simultaneously.

The procedure of executing ISDE can be described in the following:

(1) Initialization operation

Generate the initial individuals $\mathbf{x}_i^0$, $i = 1, 2, \dots, NP$ randomly in S_0 . Generate the initial antibodies $\mathbf{v}_j^0$, $j = 1, 2, \dots, m$, m denoting the number of antibodies and $m = 2NP$, randomly within $[0, 1]^2$. Generate the initial parameters τ_i^0 , $i = 1, 2, \dots, NP$, τ_{i1} and τ_{i2} denote the mutation and crossover rates, respectively, from a normal distribution ($N(0.5, \alpha)$) within $[0, 1]^2$. Determine the control parameters β , the vaccination probability P_v , and the maximal number of generations G_m . Set the current generation $G = 0$.

(2) For each individual $\mathbf{x}_i^G$, $i = 1, 2, \dots, NP$, do steps 3–5 to produce the population for the next generation $G + 1$.

(3) Mutation operation

Generate a perturbed individual $\hat{\mathbf{x}}_i^{G+1}$ using the mutation operation as

$$\hat{\mathbf{x}}_i^{G+1} = \mathbf{x}_{r_1}^G + \tau_{i1}^G \cdot (\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G). \quad (8)$$

(4) Crossover operation

Obtain the offspring $\bar{\mathbf{x}}_i^{G+1}$ from $\mathbf{x}_i^G$ and $\hat{\mathbf{x}}_i^{G+1}$ using the crossover operation as:

$$\bar{\mathbf{x}}_{it}^{G+1} = \begin{cases} \mathbf{x}_{it}^G, & P_t > \tau_{i2}^G \\ \hat{\mathbf{x}}_{it}^{G+1} & \text{otherwise,} \end{cases} \quad t = 1, 2, \dots, n. \quad (9)$$

(5) Evaluation operation

The offspring $\bar{\mathbf{x}}_i^{G+1}$ competes one-to-one with its parent $\mathbf{x}_i^G$ using the evaluation operation (7) to obtain an individual of the next generation $\mathbf{x}_i^{G+1}$.

(6) Create new antibodies

The affinity values of τ_i^G , $i = 1, 2, \dots, NP$ are calculated as:

$$A(\tau_i^G) = \begin{cases} \left| \frac{f(\mathbf{x}_i^G) - f(\mathbf{x}_i^{G+1})}{f(\mathbf{x}_i^{G+1})} \right| & \text{if } f(\mathbf{x}_i^{G+1}) < f(\mathbf{x}_i^G) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Depending on the affinity values, the parameters are ranked from best to worst. Here, we choose the first 20% parameters as the new antibodies according to the result of our trials. Then, the new antibodies, $\mathbf{v}_k^{G+1}(\text{new})$, $k = 1, 2, \dots, Q$, Q denoting the number of

the new antibodies and $Q = 0.2NP$, are selected from the top of the ranked list and the affinity value of $\nu_k^{G+1}(\text{new})$ is higher than that of $\nu_{k-1}^{G+1}(\text{new})$.

(7) Update antibodies

The antibodies in next generation ν_j^{G+1} , $j = 1, 2, \dots, m$ are updated as follows:

$$\nu_j^{G+1} = \begin{cases} \nu_{j+Q}^G & \text{if } j+Q \leq m \\ \nu_{j+Q-m}^{G+1}(\text{new}) & \text{otherwise} \end{cases} \quad (11)$$

(8) Generate the parameters for next generation

Generate the initial parameters, $\hat{\tau}_i^{G+1}$, $i = 1, 2, \dots, NP$, from a normal distribution $[N(0.5, \alpha)]$ within $[0, 1]^2$. Vaccinate the parameters with the antibodies as follows:

$$\tau_i^{G+1} = \begin{cases} \bar{\nu}_i^{G+1} & P_i \leq P_v, \\ \hat{\tau}_i^{G+1} & \text{otherwise,} \end{cases} \quad i = 1, 2, \dots, NP, \quad (12)$$

where P_i is a uniform random number in range $[0, 1]$, P_v the vaccination probability $\in [0, 1]$, and $\bar{\nu}_i^{G+1}$ is an antibody selected from ν_j^{G+1} , $j = 1, 2, \dots, m$. The probability to choose the ν_j^{G+1} as the $\bar{\nu}_i^{G+1}$ is defined as

$$P_j = \frac{j^\beta}{\sum_{j=1}^m j^\beta}, \quad (13)$$

where β is the parameter that controls the probability between different antibodies.

(9) $G = G + 1$.

(10) Repeat steps 2–9 as long as the number of generations is smaller than the allowable maximum number G_m and the best individual is not obtained.

4 BENCHMARK FUNCTION

Nine benchmark functions were used in our experimental studies. These benchmark functions were divided into three classes: functions with single optima, many local minima, and a few local minima. The benchmark functions are given in Table 1. n stands for the dimension of the function, S_0 denotes their ranges, and $f_{\min}$ is a function value of the global optimum. A more detailed description of each function is given in Yao *et al.* [30], Krink *et al.* [31], and Salman *et al.* [32].

Functions $f_1 - f_8$ (that is, Sphere function, Schwefel's problem 2.22, Step function, Rosenbrock function, Generalized Schwefel's problem 2.26, Rastrigin function, Ackley's function, and Griewank function) are high-dimensional problems. Functions f_1 , f_2 , and f_4 are unimodal. Function f_3 is the step function, which has one minimum and is discontinuous. $f_5 - f_8$ are multimodal functions where the number of local optima increases exponentially with the problem dimension. Function f_9 (Six-hump camel-back function) is a low-dimensional function with only a few local optima [30].

5 EXPERIMENTAL RESULTS

In this section, the performance of ISDE is studied with the use of the nine benchmark problems, and the obtained results are compared with those of the versions of DE proposed by Price and Storn [23], SPDE proposed by Abbass [15], and SDE proposed by

Table 1 Benchmark function

Test function	n	S_0	$f_{\min}$
$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^n$	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]^n$	0
$f_3(x) = \sum_{i=1}^n ((x_i + 0.5)^2)$	30	$[-100, 100]^n$	0
$f_4(x) = \sum_{i=1}^{n/2} [100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2]$	30	$[-30, 30]^n$	0
$f_5(x) = -\sum_{i=1}^n [x_i \sin(\sqrt{ x_i })]$	30	$[-500, 500]^n$	-12569.5
$f_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^n$	0
$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]^n$	0
$f_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 + \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	0
$f_9(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^n$	-1.03165

Omran *et al* [16]. For the original DE, $F = 0.5$ and $CR = 0.9$, as suggested by Price and Storn. For the ISDE, $P_v = 0.25$, $\alpha = 0.3$, and $\beta = 0.8$. The following parameters were used in our experiments:

Population size: $NP = 50$;

Maximal number of evaluations: 50000;

Dimensions number: 30 for $f_1 - f_8$ and 2 for f_9 .

The results reported in this section are average results of 30 independent runs.

5.1 No-noisy benchmark functions

Table 2 summarizes the results obtained by applying the different approaches to the unimodal benchmark functions. The results show that the ISDE performed better than (or at least equal to) the other strategies in all unimodal benchmark functions except the f_4 (Rosenbrock function), where the DE/best/2 and DE/rand/2 found a better solution, respectively. However, even for f_4 (Rosenbrock function), where ISDE's average is worse than DE/best/2 and DE/rand/2, it is not significantly worse. In addition, ISDE's average is better than DE/rand/1; this mutation strategy is adopted in ISDE.

Table 3 summarizes the results obtained by applying the different approaches to the multimodal benchmark functions. The results show that the ISDE significantly outperformed (or at least equal to) the other methods in all the multimodal functions.

From above experiments, it can be turned out that ISDE is clearly superior compared with the original DE strategies, SPDE, and SDE in all benchmark functions.

5.2 Noisy benchmark functions

In this subsection, the effect of noise on the performance of ISDE is investigated. The noisy versions of the benchmark functions are defined as:

$$f_{\text{noisy}}(\mathbf{x}) = f(\mathbf{x}) + N(0, 1)$$

with $N(0, 1)$ = normal (or Gaussian) distribution with mean 0 and variance 1.

Table 4 and Table 5 summarize the results obtained for the noisy problems for the unimodal and multimodal functions, respectively. Table 4 and 5 show that the ISDE was less prone to noise than other DE strategies for all benchmark functions. The ISDE retained its position as the best performer when applied to all benchmark functions even in the presence of noise. The only exception is the noisy Rastrigin's function where SDE outperformed the ISDE. However, even for the noisy Rastrigin's function where ISDE's average is worse than SDE, it is not significantly worse. In addition, the improvement is even more significant for the noisy Ackley's function, where all strategies were trapped in a local optimum. Hence, compared with the other tested strategies, the ISDE seems to be less badly affected by noise. This is a significant improvement

Table 2 Mean and standard deviation ($\pm$ SD) of the unimodal function optimization results (The data about DE/rand/1, DE/best/1, DE/rand-to-best/1, DE/rand/2, DE/best/2, SPDE, and SDE were reported by Salman *et al.* [32])

	Sphere	Schwefel problem 2.22	Rosenbrock	Step
DE/rand/1	0 $\pm$ 0	0 $\pm$ 0	32.286 $\pm$ 19.877	0 $\pm$ 0
DE/best/1	2.0969 $\pm$ 0.82958	23.242 $\pm$ 14.596	2697508 $\pm$ 1973397	5918.8 $\pm$ 2115.5
DE/rand-to-best/1	0.95313 $\pm$ 0.30027	13.032 $\pm$ 4.7294	456479 $\pm$ 375876	2142.2 $\pm$ 818.72
DE/rand/2	0 $\pm$ 0	0.050867 $\pm$ 0.017913	27.053 $\pm$ 0.94767	0 $\pm$ 0
DE/best/2	0 $\pm$ 0	0 $\pm$ 0	21.129 $\pm$ 26.826	23.167 $\pm$ 40.564
SPDE	0.000145 $\pm$ 0.000033	0.053590 $\pm$ 0.008179	697.23 $\pm$ 136.21	0 $\pm$ 0
SDE	0 $\pm$ 0	0 $\pm$ 0	52.180 $\pm$ 28.143	0 $\pm$ 0
ISDE	0 $\pm$ 0	0 $\pm$ 0	30.777 $\pm$ 16.002	0 $\pm$ 0

Table 3 Mean and standard deviation ($\pm$ SD) of the multimodal function optimization results (The data about DE/rand/1, DE/best/1, DE/rand-to-best/1, DE/rand/2, DE/best/2, SPDE, and SDE were reported by Salman *et al.* [32])

	Schwefel problem 2.26	Rastrigin	Ackley	Griewank	Camel-back
DE/rand/1	-10946 $\pm$ 1549.6	149.26 $\pm$ 35.488	0 $\pm$ 0	0.004839 $\pm$ 0.008546	-1.0316 $\pm$ 0
DE/best/1	-6911.0 $\pm$ 711.56	108.46 $\pm$ 23.937	13.704 $\pm$ 1.4439	44.928 $\pm$ 14.850	-1.0316 $\pm$ 0
DE/rand-to-best/1	-8245.2 $\pm$ 573.17	41.921 $\pm$ 9.7352	9.2557 $\pm$ 1.2586	19.949 $\pm$ 7.8826	-1.0316 $\pm$ 0
DE/rand/2	-5607.8 $\pm$ 5162.6	203.54 $\pm$ 10.761	0.0609 $\pm$ 0.001612	0.00217 $\pm$ 0.003833	-1.0316 $\pm$ 0
DE/best/2	-9358.2 $\pm$ 1458.3	172.48 $\pm$ 59.472	2.6565 $\pm$ 1.1817	0.019795 $\pm$ 0.021926	-1.0316 $\pm$ 0
SPDE	-11604 $\pm$ 235.18	60.699 $\pm$ 7.4163	0.22812 $\pm$ 0.05731	0.60925 $\pm$ 0.09809	-1.0316 $\pm$ 0
SDE	-12348 $\pm$ 198.51	5.7434 $\pm$ 2.3383	0 $\pm$ 0	0 $\pm$ 0	-1.0316 $\pm$ 0
ISDE	-12546 $\pm$ 57.352	2.0772 $\pm$ 1.7018	0 $\pm$ 0	0 $\pm$ 0	-1.0316 $\pm$ 0

Table 4 Mean and standard deviation ($\pm$ SD) of the noisy unimodal function optimization results(The data about DE/rand/1, DE/best/1, DE/rand-to-best/1, DE/rand/2, DE/best/2, SPDE, and SDE were reported by Salman *et al.* [32])

	Sphere	Schwefel problem 2.22	Rosenbrock	Step
DE/rand/1	-2.7101 ± 0.23748	-1.2760 ± 0.52326	39.809 ± 32.172	-4.1661 ± 0.31957
DE/best/1	1.3039 ± 1.6346	29.645 ± 17.440	2386294 ± 1526950	7422.5 ± 2146.2
DE/rand-to-best/1	-1.1668 ± 0.92428	15.058 ± 6.4470	4896591 ± 370528	1846.6 ± 723.14
DE/rand/2	-2.4294 ± 0.28309	0.035845 ± 0.49635	25.626 ± 0.80179	-3.9469 ± 0.34379
DE/best/2	-1.8897 ± 0.84772	2.3833 ± 4.0257	127.67 ± 117.08	4.0944 ± 6.6380
SPDE	-0.78098 ± 0.48502	3.6691 ± 0.50108	795.49 ± 177.39	-2.8130 ± 0.40006
SDE	-2.7178 ± 0.31922	-1.3804 ± 0.33657	53.318 ± 47.395	-4.1185 ± 0.28079
ISDE	-2.6558 ± 0.42365	-1.1555 ± 0.22980	32.370 ± 21.580	-2.5546 ± 0.34590

Table 5 Mean and standard deviation ($\pm$ SD) of the noisy multimodal function optimization results(The data about DE/rand/1, DE/best/1, DE/rand-to-best/1, DE/rand/2, DE/best/2, SPDE, and SDE were reported by Salman *et al.* [32])

	Schwefel problem 2.26	Rastrigin	Ackley	Griewank	Camel-back
DE/rand/1	-11048 ± 1693.8	147.23 ± 27.695	17.293 ± 0.28898	-1.7689 ± 0.29455	-5.1950 ± 0.32538
DE/best/1	-6778.5 ± 825.05	119.60 ± 22.413	14.812 ± 1.4935	64.384 ± 24.870	-5.0174 ± 0.13469
DE/rand-to-best/1	-8532.9 ± 580.32	50.859 ± 12.364	12.020 ± 2.1595	22.789 ± 7.0551	-4.9738 ± 0.36751
DE/rand/2	-4996.4 ± 3423.8	194.49 ± 14.271	17.383 ± 0.29525	-1.2637 ± 0.29686	-4.8537 ± 0.134163
DE/best/2	-9378.0 ± 1649.9	172.41 ± 38.061	15.765 ± 2.0399	0.30928 ± 1.3062	-5.1171 ± 0.29676
SPDE	-11689 ± 284.97	59.926 ± 6.5664	17.314 ± 0.35689	0.62885 ± 0.38733	-4.9702 ± 0.23305
SDE	-12304 ± 178.69	3.0424 ± 2.6218	17.191 ± 0.35062	-1.8358 ± 0.28510	-5.3876 ± 0.38736
ISDE	-12536 ± 63.415	6.9224 ± 2.8415	7.6496 ± 5.2991	-1.6548 ± 0.35933	-5.0756 ± 0.30185

over the conventional DE, which is not a good approach to achieve results with high accuracy for noisy functions [31].

5.3 Effect of α , β , and P_v

This section investigates the effect of $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, $\beta \in \{0.2, 0.4, 0.6, 0.8, 1\}$, and $P_v \in \{0.1, 0.2, 0.25, 0.3, 0.5\}$ on the performance of ISDE. Tables 6, 7, and 8 summarize the results obtained using different values of α , β , and P_v . Table 6 shows that smaller values of α are preferable for unimodal functions, whereas $\alpha = 0.3$ yields the best results for multimodal functions. In general, $\alpha = 0.3$ seems to generate the best results when applied to the all benchmark problems. Table 7 shows that the performance of the ISDE is not sensitive to the values of β , and that $\beta = 0.8$ provided the best performance for all the unimodal and multimodal functions. Table 8 shows that smaller values of P_v perform better than higher values for all the functions except for the Rastrigin's function.

6 APPLICATION

Mercury emissions from coal-fired power plants are highly dependent upon mercury speciation [33].

Mercury in the flue gas is most commonly classified in three forms: elemental mercury (Hg^0), oxidized mercury (Hg^{2+}), and particulate bound mercury (Hg^p). The particulate bound mercury is usually trapped by ash collection devices within power plants, such as electrostatic precipitators, mechanical hoppers, or bag houses. Elemental mercury is relatively inert and difficult to capture because of its nonreactivity. It is also volatile at high temperatures and insoluble in water. In contrast, oxidized mercury is very water soluble and has an affinity for adsorbing onto particulate matter such as fly ash or on metal surfaces in the duct. As a result of these physical and chemical properties of Hg^0 and Hg^{2+} , the removal of mercury is enhanced when elemental Hg is converted to its oxidized form [34].

In order to better understand the reaction mechanism that takes place in the gas phase, a model needs to be developed where the percentage of mercury oxidized can be predicted based on the concentrations of these flue gas components. The reaction mechanism, proposed by Agarwal and Stenger [34], is a five-reaction system, where two reactions are reversible and three reactions are irreversible. These reactions are listed below:

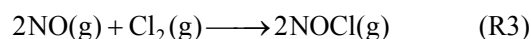
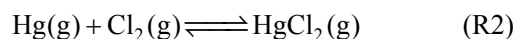
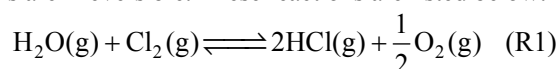


Table 6 Influence of α on the performance of ISDE

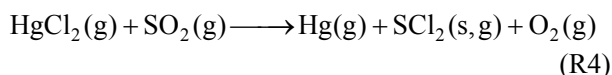
α	Sphere	Schwefel problem 2.22	Rosenbrock	Step	Schwefel problem 2.26	Rastrigin	Ackley	Griewank	Camel-back
0.1	0±0	0±0	25.079±71.534	0±0	-12569±0.018102	123.01±8.1093	0±0	0±0	-1.0316±0
0.2	0±0	0±0	29.287±14.370	0±0	-12554±40.950	37.537±6.9400	0±0	0±0	-1.0316±0
0.3	0±0	0±0	30.777±16.002	0±0	-12546±57.352	2.0772±1.7018	0±0	0±0	-1.0316±0
0.4	0±0	0±0	30.555±17.845	0±0	-12530±56.787	3.2835±1.5490	0±0	0±0	-1.0316±0
0.5	0±0	0±0	41.105±25.843	0±0	-12514±80.709	3.2834±1.5044	0±0	0±0	-1.0316±0

Table 7 Influence of β on the performance of ISDE

β	Sphere	Schwefel problem 2.22	Rosenbrock	Step	Schwefel problem 2.26	Rastrigin	Ackley	Griewank	Camel-back
0.2	0±0	0±0	35.080±21.646	0±0	-12542±59.694	3.2905±2.2134	0±0	0±0	-1.0316±0
0.4	0±0	0±0	39.045±24.352	0±0	-12538±61.686	2.3877±2.2672	0±0	0.000247±0.001350	-1.0316±0
0.6	0±0	0±0	34.543±20.804	0±0	-12550±70.127	2.2297±2.2159	0±0	0±0	-1.0316±0
0.8	0±0	0±0	30.777±16.002	0±0	-12546±57.352	2.0772±1.7018	0±0	0±0	-1.0316±0
1	0±0	0±0	34.680±22.598	0±0	-12542±57.311	2.4718±1.5707	0±0	0±0	-1.0316±0

Table 8 Influence of P_v on the performance of ISDE

P_v	Sphere	Schwefel problem 2.22	Rosenbrock	Step	Schwefel problem 2.26	Rastrigin	Ackley	Griewank	Camel-back
0.1	0±0	0±0	27.090±11.053	0±0	-12554±40.950	14.899±4.6375	0±0	0±0	-1.0316±0
0.2	0±0	0±0	30.594±18.016	0±0	-12542±59.694	5.4472±4.1039	0±0	0±0	-1.0316±0
0.25	0±0	0±0	30.777±16.002	0±0	-12546±57.352	2.0772±1.7018	0±0	0±0	-1.0316±0
0.3	0±0	0±0	32.894±18.835	0±0	-12526±58.050	3.1840±2.4706	0±0	0.000493±0.001876	-1.0316±0
0.5	0±0	0±0	53.662±28.595	0±0	-12448±157.64	9.4189±6.2966	0±0	0.000822±0.002535	-1.0316±0



where the term (s, g) denotes that the species could be in solid (s), gas (g) or both phases.

The reaction rate equations can be written as follows:

$$r_1 = A_1 \cdot \exp\left(\frac{-E_1}{RT}\right) \cdot \left([\text{H}_2\text{O}] \cdot [\text{Cl}_2] - \frac{[\text{HCl}]^2 \cdot [\text{O}_2]^{1/2}}{\exp\left(13.707 - \frac{4769.2}{T}\right)} \right) \quad (14)$$

$$r_2 = A_2 \cdot \exp\left(\frac{-E_2}{RT}\right) \cdot \left([\text{Hg}] \cdot [\text{Cl}_2] - \frac{[\text{HgCl}_2]}{\exp\left(-17.833 + \frac{17133.1}{T}\right)} \right) \quad (15)$$

$$r_3 = A_3 \cdot \exp\left(\frac{-E_3}{RT}\right) \cdot [\text{NO}] \cdot [\text{Cl}_2] \quad (16)$$

$$r_4 = A_4 \cdot \exp\left(\frac{-E_4}{RT}\right) \cdot [\text{HgCl}_2] \cdot [\text{SO}_2] \quad (17)$$

$$r_5 = A_5 \cdot \exp\left(\frac{-E_5}{RT}\right) \cdot [\text{Hg}] \cdot [\text{HCl}] \quad (18)$$

in which r_i is the reaction rate, A_i is the preexponential factor, E_i is the activation energy, $i=1, 2, 3, 4, 5$, R is the gas constant, T is absolute temperature.

The goal of this optimization problem is to determine the optimal Arrhenius parameters (A_i and E_i , $i=1, 2, 3, 4, 5$) such that the differences between the percentage of mercury oxidized calculated from Eqs. (14) through (18) and those measured experimentally is minimized. The objective function is therefore expressed by

$$\min \text{EQS} = \frac{1}{m} \sum_{i=1}^m (\bar{c}_i - c_i)^2 \quad (18)$$

in which EQS denotes the average of the squares of the differences; $\bar{c}_i$ is the Hg conversion calculated from Eqs. (14) through (18); c_i is the Hg conversion measured experimentally; m is the number of data points.

Then, ISDE was used to determine the kinetic parameters for homogeneous mercury oxidation in

Table 9 Optimal parameters and objective function values corresponding to reported data [34] and ISDE

	A_1	A_2	A_3	A_4	A_5	E_1	E_2	E_3	E_4	E_5	EQS
Ref. [34]	62.271	0.37688	98.682	138.85	36.113	8.8668	0.089337	23.509	17.638	15.108	88.134
ISDE	3.6024	0.44288	77.06	8.3261	10.866	3.9994	0.27544	17.442	13.134	13.233	73.746

flue gas with the data measured experimentally, reported by Agarwal *et al* [24, 25]. The operation parameters of ISDE were taken as $NP = 50$, $G_m = 1000$, $P_v = 0.25$, $\alpha = 0.3$, and $\beta = 0.8$. The objective function was used as a guideline to determine the best optimal parameters from the data set. The results are compared with those obtained by Agarwal and Stenger [34] in Table 9. The minimum EQS of 73.746 is obtained by ISDE. This is clearly better than reported point, $EQS = 88.134$. Hence, ISDE is found to be more profitable operating than those reported earlier in the literature.

7 CONCLUSIONS

This article presents an efficient self-adaptive DE algorithm for global optimization. In the proposed method, the immune concepts and methods are applied to determine the parameter setting, which utilizes the previous state information of search for seeking the optimal parameters, F and CR . In the benchmark tests, both noisy and nonnoisy functions, the results show that the performance of ISDE is outstanding in comparison with the original DE strategies, SPDE and SDE tested. Among the tested strategies, the ISDE can rightfully be regarded as an excellent first choice when faced with a new optimization problem to solve. Thus, the algorithm was subsequently used to estimate the kinetic parameters for homogeneous mercury oxidation in flue gas. The results marked a noticeable improvement over previously reported solutions.

REFERENCES

- Storn, R., Price, K., "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, USA (1995).
- Ingber, L., "Simulated annealing: Practice versus theory", *Math. Comput. Model.*, **18**, 29–57 (1993).
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., *Numerical Recipes in C*, Cambridge University Press, UK (1992).
- Muehlenbein, H., Schlierkamp, V., "Predictive models for the breeder genetic algorithm (I) Continuous parameter optimizations", *Evolut. Comput.*, **1**, 25–49 (1993).
- Voigt, H.M., "Soft genetic operators in evolutionary algorithms", *Lecture Notes in Computer Science*, **899**, 123–141 (1995).
- Aluffi-Pentini, F., Parisi, V., Zirilli, F., "Global optimization and stochastic differential equations", *J. Optimiz. Theory Appl.*, **47**, 1–16 (1985).
- Vesterstrom, J., Thomsen, R., "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems", In: *Proceedings of the Sixth Congress on Evolutionary Computation*, IEEE Press, USA, 332–339 (2004).
- Eberhart, R., Kennedy, J., "A new optimizer using particle swarm theory", In: *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, IEEE Press, Nagoya, Japan, 39–43 (1995).
- Krink, T., Paterlini, S., Resti, A., "Using differential evolution to improve the accuracy of bank rating systems", *Comput. Stat. Data An.*, **52**, 68–87 (2007).
- Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Harbor (1975).
- Brest, J., Bošković, B., Greiner, S., Žumer, V., Maučec, M., "Performance comparison of self-adaptive and adaptive differential evolution algorithms", *Soft Comput.*, **11**, 617–629 (2007).
- Liu, J., Lampinen, J., "A fuzzy adaptive differential revolution algorithm", In: *Proceedings of the IEEE International Region 10 Conference on Computers, Communications, Control and Power Engineering*, IEEE Press, Beijing, China, 606–611 (2002).
- Storn, R., "On the usage of differential evolution for function optimization", In: *Biennial Conference of North American Fuzzy Information Processing Society*, IEEE Press, Berkeley, USA, 519–523 (1996).
- Babu, B., Jehan, M., "Differential evolution for multi-objective optimization", In: *Proceedings of the IEEE Congress on Evolutionary Computation*, IEEE Press, Canberra, Australia, 2696–2703 (2003).
- Abbass, H., "The self-adaptive pareto differential evolution algorithm", In: *Proceedings of the IEEE Congress on Evolutionary Computation*, IEEE Press, Hawaii, USA, 831–836 (2002).
- Omran, M., Salman, A., Engelbrecht, A., "Self-adaptive differential evolution", In: *Proceedings of the International Conference on Computational Intelligence and Security*, IEEE Press, Xi'an, China, 192–199 (2005).
- Yuan, X.H., Zhang, Y., Wang, L., Yuan, Y.B., "An enhanced differential evolution algorithm for daily optimal hydro generation scheduling", *Comput. Math. Appl.*, **55**, 2458–2468 (2008).
- Nobakhti, A., Wang, H., "A simple self-adaptive differential evolution algorithm with application on the ALSTOM gasifier", *Appl. Soft Comput.*, **8**, 350–370 (2008).
- Eiben, A., Hinterding, R., Michalewicz, Z., "Parameter control in evolutionary algorithms", *IEEE T. Evolut. Comput.*, **3**, 124–141 (1999).
- Carlos, A.C.C., "Use of a self-adaptive penalty approach for engineering optimization problems", *Comput. Ind.*, **41**, 113–127 (2000).
- He, Q., Wang, L., "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Eng. Appl. Artif. Intel.*, **20**, 89–99 (2007).
- Hu, F.Z., Wang, L., He, Q., "An effective co-evolutionary differential evolution for constrained optimization", *Appl. Math. Comput.*, **186**, 340–356 (2007).
- Storn, R., Price, K., "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces", *J. Global Optim.*, **11**, 341–359 (1997).
- Agarwal, H., Stenger, H.G., Wu, S., Fan, Z., "Effects of H_2O , SO_2 and NO on homogeneous Hg oxidation by Cl_2 ", *Energ. Fuel.*, **20**,

- 1068–1075 (2006).
- 25 Agarwal, H., Romero, C.E., Stenger, H.G., “Comparing and interpreting laboratory results of Hg oxidation by a chlorine species”, *Fuel Process. Technol.*, **88**, 723–730 (2007).
- 26 Farmer, J.D., Packard, N.H., Perelson, A.S., “The immune system, adaptation, and machine learning”, *Phys. D*, **2**, 187–204 (1986).
- 27 Wu, X.L., Lu, J.G., Sun, Y.X., “An improved differential evolution for optimization of chemical process”, *Chin. J. Chem. Eng.*, **16**, 228–234 (2008).
- 28 Jiao, L.C., Wang, L., “A novel genetic algorithm based on immunity”, *IEEE T. Syst. Man Cy. A*, **30**, 552–561 (2000).
- 29 Zeng, C.W., Gu, T.L., “A novel immunity-growth genetic algorithm for traveling salesman problem”, In: Proceedings of IEEE Conference on Natural Computation, IEEE Press, Haikou, China, 394–398 (2007).
- 30 Xin, Y., Liu, Y., Lin, G.M., “Evolutionary programming made faster”, *IEEE T. Evolut. Comput.*, **3**, 82–102 (1999).
- 31 Krink, T., Filipič, B., Fogel, B., “Noisy optimization problems—A particular challenge for differential evolution?”, In: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE Press, Portland, USA, 332–339 (2004).
- 32 Salman, A., Engelbrecht, A., Omran, M., “Empirical analysis of self-adaptive differential evolution”, *Eur. J. Oper. Res.*, **83**, 785–804 (2007).
- 33 Niksa, S., Helble, J.J., Fujiwara, N., “Kinetic modeling of homogeneous mercury oxidation: The importance of NO and H₂O in predicting oxidation in coal-derived systems”, *Environ. Sci. Technol.*, **35**, 3701–3706 (2001).
- 34 Agarwal, H., Stenger, H.G., “Development of a predictive kinetic model for homogeneous Hg oxidation data”, *Math. Comput. Model.*, **45**, 109–125 (2007).