



Evaluation of integrated differential evolution and unified bare-bones particle swarm optimization for phase equilibrium and stability problems

Haibo Zhang^a, Jorge Adán Fernández-Vargas^b, Gade Pandu Rangaiah^{a,*},
Adrián Bonilla-Petriciolet^c, Juan Gabriel Segovia-Hernández^b

^a Department of Chemical & Biomolecular Engineering, Engineering Drive 4, National University of Singapore, Singapore 117576, Singapore

^b Department of Chemical Engineering, Universidad de Guanajuato, México 36050, Mexico

^c Department of Chemical Engineering, Instituto Tecnológico de Aguascalientes, México 20256, Mexico

ARTICLE INFO

Article history:

Received 8 March 2011

Received in revised form 2 August 2011

Accepted 3 August 2011

Available online 10 August 2011

Keywords:

Integrated differential evolution

Bare-bones particle swarm optimization

Phase equilibrium calculations

Chemical equilibrium calculations

Phase stability analysis

ABSTRACT

Phase equilibrium calculations and phase stability analysis play a significant role in the simulation, design and optimization of separation processes in chemical engineering. These are very challenging problems due to the high non-linearity of thermodynamic models. Global optimization methods are required in order to solve these complex, non-convex optimization problems. Recently, stochastic global optimization algorithms were applied to solve these problems. However, these optimization methods have some parameters that need to be tuned in order to obtain good reliability and efficiency. In this study, we introduce three global optimization algorithms developed by our group for phase and chemical equilibrium calculations, namely, unified bare-bones particle swarm optimization (UBBPSO), integrated differential evolution (IDE) and IDE without tabu list and radius (IDE.N), which have fewer control parameters to be tuned. The performance of these three stochastic algorithms is tested and compared in order to identify their relative strengths for phase equilibrium and phase stability problems. The phase equilibrium problems include both without and with chemical reactions. Our results show that the effectiveness of the stochastic methods tested depends on the stopping criterion. Overall, IDE has achieved better performance for the phase equilibrium, chemical equilibrium and phase stability problems.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Phase equilibrium calculations (PEC) and phase stability (PS) problems are crucial during the analysis of chemical process. Novel processes handle complex mixtures, severe operating conditions, or even incorporate combined unit operations (e.g., reactive distillation, extractive distillation, etc.). The reliable computation of the thermodynamic state for these systems is especially important due to the direct impact of wrong estimations on energy consumption and operating costs. When a mixture is analyzed, PEC involves not only the calculation of number of moles of each phase but also the number of stable phases where PS is used to determine the stability of the calculated composition at equilibrium. In general, number and type of phases, at which Gibbs free energy function achieves the global minimum, are unknown in PEC problems, and so several calculations may have to be performed using different phase configurations to identify the stable equilibrium state. Therefore, the unknown phases of general PEC increase the complexity of the

optimization problem. Both PEC and PS problems require the global optimization of a specific function; usually, these have to be solved many times during a simulation. Specifically, PS analysis requires the minimization of tangent plane distance function (TPDF), while the Gibbs free energy function needs to be minimized for PEC [1]. A reactive phase equilibrium calculation (rPEC) or chemical equilibrium, is performed if any reaction is possible in the system under study, and the objective function must satisfy the chemical equilibrium constraints.

In general, there are several challenges in finding the global optimum of Gibbs free energy function. First, number and type of phases where the thermodynamic function achieves the global optimum are usually unknown *a priori*. Second, high non-linearity of thermodynamic models, non-convexity of Gibbs free energy function and the presence of a trivial solution in the search space make PEC and PS problems difficult to solve. Moreover, for a fixed number of phases and components, Gibbs free energy function may have local optimal values that are very comparable to the global optimum value, which makes it challenging to find the global optimum [2]. Thus, PEC, rPEC and PS problems require a reliable, efficient and robust global optimization algorithm. Further, application of global optimization techniques to these problems is very challenging.

* Corresponding author. Tel.: +65 65162187; fax: +65 67791936.

E-mail address: chegepr@nus.edu.sg (G.P. Rangaiah).

Many deterministic and stochastic optimization algorithms have been proposed and tested for finding the global optimum in PEC, rPEC and PS problems, particularly in the past two decades [2–11]. Deterministic global optimization studies have been applied to different PEC, PS and/or rPEC problems. Homotopy continuation methods have been applied to PEC and PS problems [8,12]. Although homotopy-continuation algorithm guarantees global convergence to a single solution, it does not guarantee global convergence to multiple solutions. Even using complex search spaces, the success of continuation methods in finding all solutions cannot be assured. Burgos-Solorzano et al. [11] applied interval Newton method for solving the PEC problems under high pressure. This method can solve non-linear equations to find all solutions lying within the variable bounds. It requires an interval extension of the Jacobian matrix, and involves setting up and solving the interval Newton equation for a new interval. However, it is very hard to find all solutions and Jacobian matrix for the complex systems, and the computational time is significant for multi-component systems.

Recently, Rossi et al. [13] applied convex analysis method to PEC and rPEC problems. This method employs the CONOPT solver in GAMS (general algebraic modeling system). The proposed method can solve PEC problems with high efficiency and reliability but it requires the convexity of the model. Branch and bound methods have been applied to many applications including PS and PEC problems [14,15]. In general, these methods are often slow and require a significant numerical effort that grows exponentially with problem size [16,17]. Besides, branch and bound methods require certain properties of the objective function, and problem reformulation is usually needed to guarantee the global convergence. Note that the problem reformulation can be very difficult to perform, especially for complex thermodynamic models such as equations of state with non-traditional mixing rules. Finally, Nichita et al. applied the tunneling method to perform stability analysis of various systems [18,19] and to PEC problems [17,20]. Their results suggest that tunneling method is a robust and efficient tool for these applications even for difficult cases. However, it requires feasible and improved initial estimates for reliability and computational efficiency, respectively [17]. For an unknown system, it is very difficult to provide a feasible and good initial estimate for the algorithm.

In summary, the deterministic methods can guarantee convergence to the global optimum but they usually require certain properties such as continuity, *a priori* information of the system; reformulation of the problem may be needed depending on the characteristic of the thermodynamic models, and the computational time grows exponentially with problem size. In contrast, stochastic methods are quite simple to implement and use. They do not require any assumptions or transformation of the original problems, can be applied with any thermodynamic model, and yet provide a high probabilistic convergence to the global optimum. They can often locate the global optimum in modest computational time compared to deterministic methods [2].

In recent years, several stochastic global optimization techniques have been applied to solve the PS and PEC problems in non-reactive and reactive systems [1–6,17–22]. These algorithms include simulated annealing (SA), genetic algorithms (GA), tabu search (TS), differential evolution (DE), random tunneling method (RT) and particle swarm optimization (PSO). In particular, Srinivas and Rangaiah [3,23] studied DE and TS for non-reactive mixtures, and proposed two versions of DE with tabu list (DETL), in order to improve the performance of the optimization algorithm. Srinivas and Rangaiah [22] evaluated the RT on a number of medium sized problems including vapor–liquid, liquid–liquid and vapor–liquid–liquid equilibrium problems. RT can locate the global optimum for most of the examples tested but its reliability is low for problems having a local minimum comparable to the global mini-

mum. In a recent study, Bonilla-Petriciolet and Segovia-Hernández [6] tested different versions of PSO for PS and PEC for both reactive and non-reactive systems, and their results show that classical PSO is a reliable method with good performance.

Systematic and comprehensive comparison of different global optimization methods is challenging. However, some comparison of stochastic with deterministic algorithms for phase equilibrium calculations can be found in the literature. Teh and Rangaiah [24,25] compared GA and TS with several deterministic algorithms such as Rachford–Rice–mean value theorem–Wegstein's projection method, Accelerated successive substitution method, Nelson's method, simultaneous equation-solving method, linearly constrained minimization method, GLOPEQ and enhanced interval analysis method for solving phase equilibrium calculations. Their comparison shows that some stochastic methods can be more efficient than deterministic algorithms.

Most of the stochastic methods have some parameters to be tuned for different problems in order to improve the convergence to the global optimum. Selection of proper parameter values for different problems usually cost a lot of effort, and an improper choice can result in computational inefficiency or poor reliability. In order to overcome such difficulties, this work evaluates three global optimization algorithms (unified bare-bones particle swarm optimization, UBBPSO; integrated differential evolution, IDE; IDE without tabu list and radius, IDE.N) that have fewer algorithm parameters, for PEC, rPEC and PS problems involving multiple components, multiple phases and popular thermodynamic models. The performance of UBBPSO, IDE and IDE.N on PEC, rPEC and PS problems are compared and discussed based on both reliability and computational efficiency using practical stopping criteria.

The rest of this paper is organized as follows. The classical PSO and UBBPSO algorithms are described in Section 2. The IDE algorithm is described in Section 3. Description of PEC, PS and rPEC problems is given in Section 4. Implementation of UBBPSO, IDE.N and IDE is presented in Section 5. Section 6 presents the results and discusses the performance of UBBPSO, IDE.N and IDE on PEC, PS and rPEC problems. Finally, the conclusions of this work are summarized in Section 7.

2. Description of classical PSO and UBBPSO methods

In this study, the global optimization problem to be solved is:

$$\text{Minimize } F(X) \quad (1)$$

with respect to D decision variables: $X = (X^1, X^2, \dots, X^d, \dots, X^D)$. The upper and lower bounds of these variables are $X_{\max}^1, X_{\max}^2, \dots, X_{\max}^d, \dots, X_{\max}^D$ and $X_{\min}^1, X_{\min}^2, \dots, X_{\min}^d, \dots, X_{\min}^D$, respectively.

2.1. Classical PSO

PSO is one of the population-based stochastic global optimization technique proposed by Eberhart and Kennedy [13] based on the social behavior of animals such as bird flocking, fish schooling and swarm theory. Every particle in the population represents a point in the D -dimensional search space, and serves as a potential solution. The movement of each particle is guided by its respective velocity which is influenced by the personal best position (*pbest*) found so far by the particle and also by the neighborhood best position (*nbest*) found so far by neighboring particles. Note that *nbest* can include *gbest* (global best) and *lbest* (local best). Even though PSO has many similarities with evolutionary computation techniques, evolution operators such as crossover and mutation are not implemented in its algorithm.

The classical PSO algorithm [26] updates the particles in the population by:

$$V_i^{d,k+1} = V_i^{d,k} + c_1 \text{Rand}1_i^d(pbest_i^{d,k} - X_i^{d,k}) + c_2 \text{Rand}2_i^d(gbest^d - X_i^{d,k}) \quad (2)$$

$$X_i^{d,k+1} = X_i^{d,k} + V_i^{d,k+1} \quad (3)$$

Here, $V_i^{d,k}$ and $X_i^{d,k}$ are, respectively, d th element of velocity and position vectors of i th particle at k th iteration, and $\text{Rand}1$ and $\text{Rand}2$ are two uniformly distributed random numbers in the range $[0,1]$, which are different for each dimension of each particle in each iteration, as indicated by the superscript and subscript. In Eq. (2), c_1 and c_2 are the weights for the stochastic acceleration used to adjust the movement of the particle's dimension towards the corresponding $pbest$ and $gbest$ dimensions. Thus, they are commonly termed as the acceleration coefficients; they are also known as learning factors (i.e., cognitive and social parameters) towards the best positions [27].

2.2. Development of unified BBPSO

In order to eliminate the parameters in classical PSO and to have an adaptive balance between exploration and exploitation, the dynamic adaptation of parameters is applied. Specifically, a particle can be updated using Gaussian normal distribution with mean and standard deviation given by

$$\mu_i^{d,k} = \frac{(pbest_i^{d,k} + gbest^d)}{2} \quad (4)$$

$$\sigma_i^{d,k} = |pbest_i^{d,k} - gbest^d| \quad (5)$$

$$X_i^{d,k+1} = \begin{cases} N(\mu_i^{d,k}, \sigma_i^{d,k}) & \text{if } U[0, 1] < 0.5 \\ pbest_i^{d,k} & \text{otherwise} \end{cases} \quad (6)$$

Here, $\mu_i^{d,k}$ and $\sigma_i^{d,k}$ are the mean and absolute difference of $pbest$ and $gbest$, respectively, in d th dimension of the i th particle at k th iteration. In Eq. (6), N represents Gaussian normal distribution with mean $\mu_i^{d,k}$ and standard deviation $\sigma_i^{d,k}$. The use of Gaussian normal distribution to update the particle is known as BBPSO [28]. However, the algorithm still faces some difficulties such as premature convergence, and can be trapped easily at local optimum. For example, if the current particle's $pbest$ objective function value happens to be the same as that of $nbest$ (which can be $gbest$ or $lbest$ depending on the topology employed), then the mean in Eq. (6) is the value of the particle's $pbest$ position itself and the standard deviation is zero. In such a case, the $nbest$ -index particle (i.e., the particle whose $pbest$ is the same as $nbest$) will not be updated.

In order to overcome the premature convergence of BBPSO, Zhang et al. [29] proposed a BBPSO with crossover and mutation operators of DE based on $gbest$ and $lbest$ neighborhood topology, namely, BBPSO-MC- $gbest$ and BBPSO-MC- $lbest$. In both these algorithms, mutation and crossover operators of DE algorithm are employed for updating only the $nbest$ -index particle by:

$$X_i^{d,k+1} = \begin{cases} pbest_{i_1}^{d,k} + 0.5 \times (pbest_{i_2}^{d,k} - pbest_{i_3}^{d,k}) & \text{if } U[0, 1] < 0.5 \\ nbest_i^d & \text{Otherwise} \end{cases}$$

where $i \neq i_1 \neq i_2 \neq i_3$, which means $pbest$'s in the above equation are randomly chosen. Hence, there is 50% chance that the $nbest$ -index particle is updated by the mutation operator of DE, whose information comes from the randomly chosen $pbest$'s other than the current particle's $pbest$. In this case, the mutation and crossover rates of 0.5 each are implemented without any selection operation

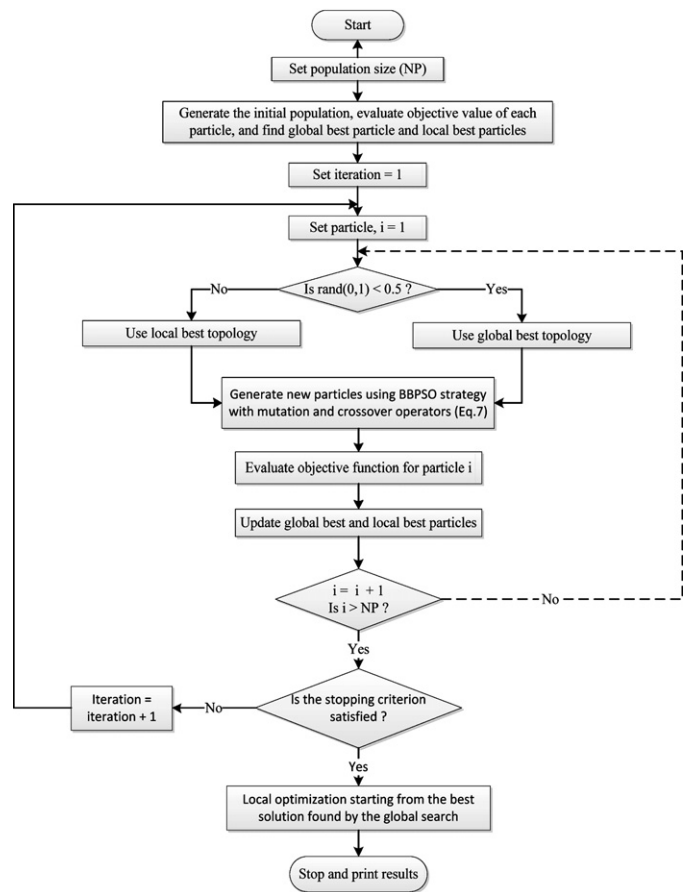


Fig. 1. Flowchart of UBBPSO algorithm.

of DE. This approach enables the $nbest$ -index particle to retain the good variables and has a possibility of changing the non-performing variables. The two BBPSO-MC algorithms have been tested on a number of benchmark functions and applied to parameter estimation in vapor–liquid equilibrium data modeling problems [29]. The results show that BBPSO-MC- $gbest$ and BBPSO-MC- $lbest$ have their own strong points. In order to combine the strong features of the BBPSO-MC- $gbest$ and BBPSO-MC- $lbest$, here we combine both versions into UBBPSO by using 50% probability for $gbest$ and $lbest$ topologies. The flowchart of UBBPSO is shown in Fig. 1; towards the end, a local optimizer is used to refine the best solution found by the global search.

3. Description of IDE method

DE is another population-based stochastic global optimization algorithm proposed by Storn and Price [30]. It has been gaining popularity in the recent past due to its simplicity and capability to handle non-convex and non-differentiable objective functions.

$$\text{for } i = 1, 2, \dots, D \quad (7)$$

Moreover, DE has relatively faster convergence and high reliability to find the global optimum [1,31]. Its main steps are initialization, mutation, crossover and selection, whose details can be found in [30]. In the recent past, DE has been successfully applied to

Step 1: Initialization

Initialize the generation counter, $G = 0$. Randomly initialize NP individuals (target vectors, $X_{i,G} = \{x_{i,G}^1, \dots, x_{i,G}^D\}$ for $i = 1, 2, \dots, NP$) within the search range $[X_{min}, X_{max}]$, where $X_{min} = \{x_{min}^1, \dots, x_{min}^D\}$ and $X_{max} = \{x_{max}^1, \dots, x_{max}^D\}$.

FOR $i = 1$ to NP

FOR $j = 1$ to D

$x_{i,G}^j = x_{min}^j + rand(0, 1)(x_{max}^j - x_{min}^j)$

END FOR

END FOR

Evaluate all the individuals and send them to tabu list

Set **gbest** equal to the global best individual

Step 2: Generation, G

WHILE the stopping criterion is not satisfied

 Calculate the mutation strategy probabilities, $P_{k,G}$ based on the previous generations

 Use stochastic universal sampling to assign mutation strategy for each individual based on $P_{k,G}$

FOR $k = 1$ to K / where K is the total number of mutation strategies/

$CRm_k = \text{median}(CRMemory_k)$

FOR $i = 1$ to NP

$CR_{k,i} = \text{Normrnd}(CRm_k, 0.1)$ /Assign crossover rate for each individual i /

END FOR

END FOR

FOR $i = 1$ to NP

DO

$F_i = \text{Normrnd}(0.5, 0.3)$ /Assign mutation factor/

 Mutation /Generate a mutation vector, $V_{i,G} = \{v_{i,G}^1, \dots, v_{i,G}^D\}$ /

 Crossover /Generate a trial vector, $U_{i,G} = \{u_{i,G}^1, \dots, u_{i,G}^D\}$ /

 Check boundary violations of the trial vector and correct them

UNTIL the tabu check is satisfied (Euclidean distance $> TR$)

 Evaluate the trial vector and send it to tabu list

IF $F(U_{i,G}) \leq F(X_{i,G})$ /Selection/

$X_{i,G} = U_{i,G}; F(X_{i,G+1}) = F(U_{i,G})$

$ns_{k,G} = ns_{k,G} + 1$; Store $CR_{k,i}$ into $CRMemory_k$

IF $F(U_{i,G}) \leq F(\mathbf{gbest})$

$\mathbf{gbest} = U_{i,G}; F(\mathbf{gbest}) = F(U_{i,G})$

END IF

ELSE $nf_{k,G} = nf_{k,G} + 1$

END IF

END FOR

$G = G + 1$

END WHILE

Local optimization starting from the **gbest** found by the global search

Output: **Solution found by local optimizer**

Fig. 2. Pseudo-code of IDE algorithm.

diverse fields, e.g., phase equilibrium, phase stability problems and parameter estimation problems [1–3], fed-batch bioreactor [32], synthesis of cost-optimal heat exchange networks [33], etc.

In DE, users need to choose suitable values of parameters (such as scaling factor F and crossover rate Cr), and also proper mutation strategy for different problems in order to enhance convergence to the global optimum. These selections usually require lot of effort, and an improper choice can result in computational inefficacy. In order to overcome such problems, researchers have been studying strategies to adapt the parameters of DE [34–36]. Recently, Zhang and Rangaiah [37] proposed IDE by integrating the strong features of tabu search, self-adaptive strategy of [23] and a local optimizer. The tabu list of tabu search can prevent revisiting the same area, the self-adaptive strategy can tune the parameters and mutation strategies based on the learning experience of previous generations, and the local optimizer efficiently improves the accuracy of the best point found by the global algorithm.

The pseudo-code of IDE algorithm is presented in Fig. 2. The algorithm generates the initial population of NP individuals using uniformly distributed random numbers within the search space.

The objective function of each individual is evaluated and sent to the tabu list. Generation counter, G is initialized to 1, probability of each mutation strategy, $P_{k,G}$ to 0.25 and median value of crossover rate for each strategy, $Cr_{m,k}$ to 0.5. $P_{k,G}$ is updated based on the learning experience from the previous generations. The median of crossover rate values ($Cr_{i,k}$) stored inside $CRMemory_k$ is used for $Cr_{m,k}$. During each generation, a strategy for each target individual is selected with probability $P_{k,G}$ using stochastic universal selection method [38].

The $Cr_{i,k}$ for each trial individual is calculated based on normal distribution with mean of $Cr_{m,k}$ and standard deviation of 0.1. For generating a new trial individual, mutation factor F is calculated based on normal distribution with mean of 0.5 and standard deviation of 0.3, and then a new trial individual is produced according to assigned mutation strategy, F and $Cr_{i,k}$. A boundary violation check is performed to make sure the decision variables of the new trial individual is within the search space. If any bound is violated, the corresponding decision variable of the trial individual is replaced by a randomly generated value within its bounds. The trial individual is then compared with the points in the taboo list. If it is

near to any point in the tabu list, the trial individual is rejected because it may not bring any new information about the objective function and only increases the number of function evaluations. The rejected individual is replaced by generating another trial individual through the mutation and crossover operations. Else, the algorithm continues as follows.

After evaluating the objective function of the trial individual generated, the selection step is performed based on the fitness of the objective function value and the tabu list is updated dynamically to keep the latest points in the list. If the trial individual is selected, it replaces the target individual in the population immediately, and $Cr_{i,k}$ is stored into Crm_k and the success of the corresponding mutation strategy is updated. The above steps are repeated for all NP target individuals in the population, which completes one generation. The updating of P_{kG} , calculation of Crm_k , mutation, crossover and selection operations are repeated for the next generation until the stopping criterion is satisfied. Then, the best point obtained over all generations is refined using the local optimizer.

The IDE algorithm without tabu list (TL) and tabu radius (TR), which is named as IDE.N, is applied to PEC, rPEC and PS problems to assess the benefits of using tabu concept. The generation steps of IDE.N are exactly same with IDE (Fig. 2) except the absence of tabu list and tabu check operations.

4. Description of PEC, PS and rPEC problems

A brief description of the global optimization problems including the objective function, decision variables and constraints, for PEC, PS and rPEC problems is given in the following sections.

4.1. Description of PEC problems

A mixture of substances at a given temperature, T , pressure, P and total molar amount may separate into two or more phases. The composition of the different substances is the same throughout a phase but may significantly vary in different phases at equilibrium. If there is no reaction between the different substances, then it is a phase equilibrium problem. There are mainly two different approaches for PEC: equation solving approach and Gibbs free energy minimization approach. The former involves solving a set of non-linear equations arising from mass balances and equilibrium relationships. The latter involves the minimization of the Gibbs free energy function. Although the first approach seems to be faster and simple, the solution obtained may not correspond to the global minimum of Gibbs free energy function. Moreover, it needs a priori knowledge of phases existing at equilibrium [5]. Classic thermodynamics indicate that minimization of Gibbs free energy is a natural approach for calculating the equilibrium state of a mixture. Hence, this study uses Gibbs free energy minimization for PEC, which was used to determine phase compositions at equilibrium in several works [4,5,25,39].

The mathematical formulation involves the minimization of a non-convex objective function (Gibbs free energy) subject to mass balance equality constraints and bounds that limit the range of variables. In a non-reactive system with c components and π phases, the objective function for PEC is

$$g = \sum_{j=1}^{\pi} \sum_{i=1}^c n_{ij} \ln(x_{ij} \gamma_{ij}) = \sum_{j=1}^{\pi} \sum_{i=1}^c n_{ij} \ln \left(\frac{x_{ij} \hat{\phi}_{ij}}{\phi_i} \right) \quad (8)$$

where n_{ij} , x_{ij} , γ_{ij} and $\hat{\phi}_{ij}$ are, respectively, the moles, mole fraction, activity coefficient and fugacity coefficient of component i in phase j , and ϕ_i is the fugacity coefficient of pure component. Eq. (8) must

be minimized with respect to n_{ij} taking into account the following mass balance constraints:

$$\sum_{j=1}^{\pi} n_{ij} = z_i n_F \quad i = 1, \dots, c \quad (9)$$

$$0 \leq n_{ij} \leq z_i n_F \quad i = 1, \dots, c \quad j = 1, \dots, \pi \quad (10)$$

where z_i is the mole fraction of component i in the feed and n_F is the total moles in the feed.

To perform unconstrained minimization of Gibbs energy function, we can use new variables instead of n_{ij} as decision variables. The introduction of the new variables eliminates the restrictions imposed by material balances, reduces problem dimensionality and the optimization problem is transformed into an unconstrained one. For multi-phase non-reactive systems, new variables $\beta_{ij} \in (0,1)$ are defined and employed as decision variables by using the following expressions:

$$n_{i1} = \beta_{i1} z_i n_F \quad i = 1, \dots, c \quad (11)$$

$$n_{ij} = \beta_{ij} \left(z_i n_F - \sum_{m=1}^{j-1} n_{im} \right) \quad i = 1, \dots, c; \quad j = 2, \dots, \pi - 1 \quad (12)$$

$$n_{i\pi} = z_i n_F - \sum_{m=1}^{\pi-1} n_{im} \quad i = 1, \dots, c \quad (13)$$

Using this formulation, all trial compositions satisfy the mass balances allowing the easy application of optimization strategies [3,6]. For Gibbs energy minimization, the number of phases existing at the equilibrium is assumed to be known *a priori*, and the number of decision variables β_{ij} is $c(\pi - 1)$ for non-reactive systems.

Details of PEC problems used in this study are in Table 1. In most of the reported studies, PEC problems tested are assuming that the number and type of phases are known; such problems are also known as phase split calculations. In this study too, the same assumption is made, and the problems tested are simply referred to as PEC problems.

4.2. Description of phase stability problems

Phase stability (PS) problem is used to determine the thermodynamic state that corresponds to the global minimum of Gibbs free energy. Its results can be used to find good starting points to improve the reliability of PEC. PS is often tested using the tangent plane criterion, which states that a phase is stable provided that the tangent plane generated at the corresponding composition lies below the molar Gibbs energy surface for all compositions [12,14]. As an alternative, Mitsos and Barton [40] reinterpreted the Gibbs tangent plane stability criterion via a Lagrangian duality approach, as the solution of the dual problem of a primal problem that minimizes Gibbs free energy subject to material balances for solving the PS problems.

One common implementation of the tangent plane criterion [12,14] is to minimize the tangent plane distance function (TPDF), defined as the vertical distance between the molar Gibbs energy surface and the tangent plane at the given phase composition. Specifically, TPDF is given by

$$\text{TPDF} = \sum_{i=1}^c y_i (\mu_i|_y - \mu_i|_z) \quad (14)$$

Table 1
Details of PEC and PS problems studied.

PEC and PS No.	System	Feed conditions	Thermodynamic models	Global optimum for	
				Equilibrium	Stability
1	n-Butyl acetate + water	$n_F = (0.5, 0.5)$ at 298 K and 101.325 kPa	NRTL model and parameters reported by Rangaiah [5].	−0.020198	−0.032466
2	Toluene + water + aniline	$n_F = (0.29989, 0.20006, 0.50005)$ at 298 K and 101.325 kPa	NRTL model and Model parameters reported by McDonald and Floudas [39].	−0.352957	−0.294540
3	$N_2 + C_1 + C_2$	$n_F = (0.3, 0.1, 0.6)$ at 270 K and 7600 kPa	SRK EoS with classical mixing rules. Model parameters reported by Bonilla-Petriciolet et al. [21].	−0.547791	−0.015767
4	$C_1 + H_2S$	$n_F = (0.9813, 0.0187)$ at 190 K and 4053 kPa	SRK EoS with classical mixing rules. Model parameters reported by Rangaiah [5].	−0.019892	−0.003932
5	$C_2 + C_3 + C_4 + C_5 + C_6$	$n_F = (0.401, 0.293, 0.199, 0.0707, 0.0363)$ at 390 K and 5583 kPa	SRK EoS with classical mixing rules. Model parameters reported by Bonilla-Petriciolet et al. [21].	−1.183653	−0.000002
6	$C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_{7-16} + C_{17+}$	$n_F = (0.7212, 0.09205, 0.04455, 0.03123, 0.01273, 0.01361, 0.07215, 0.01248)$ at 353 K and 38500 kPa	SRK EoS with classical mixing rules. Model parameters reported by Harding and Floudas [14].	−0.838783	−0.002688
7	$C_1 + C_2 + C_3 + iC_4 + C_4 + iC_5 + C_5 + C_6 + iC_{15}$	$n_F = (0.614, 0.10259, 0.04985, 0.008989, 0.02116, 0.00722, 0.01187, 0.01435, 0.16998)$ at 314 K and 2010.288 kPa	SRK EoS with classical mixing rules. Model parameters reported by Rangaiah [5].	−0.769772	−1.486205
8	$C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_7 + C_8 + C_9 + C_{10}$	$n_F = (0.6436, 0.0752, 0.0474, 0.0412, 0.0297, 0.0138, 0.0303, 0.0371, 0.0415, 0.0402)$ at 435.35 K and 19150 kPa	SRK EoS with classical mixing rules. Model parameters reported by Bonilla-Petriciolet et al. [21].	−1.121176	−0.0000205

where $\mu_i|_y$ and $\mu_i|_z$ are the chemical potentials of component i calculated at compositions y and z , respectively. Eq. (14) is the objective function, and the constraint and bounds are

$$\sum_{i=1}^c y_i = 1 \text{ and } 0 \leq y_i \leq 1 \quad (15)$$

For stability analysis of a phase/mixture of composition z , TPDF must be globally minimized with respect to composition of a trial phase y . If the global minimum value of TPDF is zero, then the specified phase and others sharing the same tangent plane would coexist at equilibrium. The decision variables in phase stability problems are y_i for $i = 1, 2, \dots, c$.

The constrained global optimization of TPDF can be transformed into an unconstrained problem by using decision variables β_i instead of y_i as follows:

$$n_{iy} = \beta_i z_i n_F \quad i = 1, \dots, c \quad (16)$$

and

$$y_i = \frac{n_{iy}}{\sum_{j=1}^c n_{iy}} \quad i = 1, \dots, c \quad (17)$$

where n_F is the total moles in the feed mixture used for stability analysis, and n_{iy} are the conventional mole numbers of component i in trial phase y . The number of decision variables is still c for the unconstrained minimization of TPDF. Thus, the unconstrained global optimization problem for phase stability analysis is:

$$\begin{aligned} \min \quad & \text{TPDF}(\beta) \\ \text{s.t.} \quad & 0 \leq \beta_i \leq 1 \quad i = 1, \dots, c \end{aligned} \quad (18)$$

The calculation of TPDF is straightforward with almost any thermodynamic model because:

$$\frac{\mu_i - \mu_i^0}{R_g T} = \ln \left(\frac{x_i \hat{\phi}_i}{\phi_i} \right) = \ln(x_i \gamma_i) \quad (19)$$

where R_g is the universal gas constant, μ_i is the chemical potential of component i at the mixture, and μ_i^0 is the chemical potential of pure component i . More details of PS problem formulation can be found in [5]. Characteristics of PS problems used in this study are summarized in Table 1.

4.3. Description of rPEC or (chemical equilibrium) problems

In rPEC problems, also known as chemical equilibrium problems, reactions increase the complexity and dimensionality of phase equilibrium problems, and so phase split calculations in reactive systems are more challenging due to non-linear interactions among phases and reactions. The phase distribution and composition at equilibrium of a reactive mixture are determined by the global minimization of Gibbs free energy subject to element/mass balances and chemical equilibrium constraints [11,41]. Therefore, to determine the phase equilibrium compositions in reactive systems, it is necessary to find the global minimum of the free energy with respect to mole numbers of components in each of the phases subject to constraints and bounds. The expressions for Gibbs free energy and its mathematical properties depend on the structure of the thermodynamic equation(s) chosen to model each of the phases that may exist at equilibrium [2].

Recently, Bonilla-Petriciolet et al. [2] concluded that the constrained Gibbs free energy minimization approach has the advantage of requiring smaller computing time compared to the unconstrained approach, is straightforward and suitable for chemical equilibrium calculations. In summary, for a system with c components and π phases subject to r independent chemical reactions, the constrained objective function for rPEC is

$$F_{\text{obj}} = g - \sum_{j=1}^{\pi} \ln K_{\text{eq}} N^{-1} n_{\text{ref},j} \quad (20)$$

where g is given by Eq. (8), $\ln K_{\text{eq}}$ is a row vector of logarithms of chemical equilibrium constants for r independent reactions, N is

an invertible, square matrix formed from the stoichiometric coefficients of a set of reference components chosen from r reactions, and n_{ref} is a column vector of moles of each of the reference components. This objective function is defined using reaction equilibrium constants, and it must be globally minimized subject to the following mass balance restrictions [2]:

$$\sum_{j=1}^{\pi} (n_{ij} - v_i N^{-1} n_{\text{ref},j}) = n_{iF} - v_i N^{-1} n_{\text{ref},F} \quad i = 1, \dots, c - r \quad (21)$$

where n_{iF} is the initial moles of component i in the feed. These mass balance equations can be rearranged to reduce the number of decision variables of the optimization problem and to eliminate equality constraints, which are usually challenging for stochastic optimization methods. Thus, Eq. (21) is rearranged to reduce the number of decision variables using the following expression:

$$n_{i\pi} = n_{iF} - v_i N^{-1} (n_{\text{ref},F} - n_{\text{ref},\pi}) - \sum_{j=1}^{\pi-1} (n_{ij} - v_i N^{-1} n_{\text{ref},j}) \quad i = 1, \dots, c - r \quad (22)$$

Using Eq. (22), the decision variables for rPEC are $c(\pi - 1) + r$ mole numbers (n_{ij}). Then, the global optimization problem can be solved by minimizing Eq. (20) with respect to $c(\pi - 1) + r$ decision variables n_{ij} and the remaining $c - r$ mole numbers ($n_{i\pi}$) are determined from Eq. (22), subject to the inequality constraints $n_{i\pi} > 0$.

In constrained optimization problems, the search space consists of both feasible and infeasible points. For rPEC, feasible points satisfy all the mass balance constraints, Eq. (21), while infeasible points violate at least one of them (i.e., $n_{i\pi} < 0$ where $i = 1, \dots, c - r$). The penalty function method is used to solve the constrained Gibbs

free energy minimization in reactive systems because it is easy to implement and is considered efficient for handling constraints in the stochastic methods [2]. For handling these constraints, an absolute value of constraint violation is multiplied with a high penalty weight and then added to the objective function. In case of more than one constraint violation, all constraint violations are first multiplied with the penalty weight, and all of them are added to the objective function. Specifically, the penalty function is given by

$$F_r = \begin{cases} F_{\text{obj}} & \text{if } \forall n_{ij} > 0 \quad i = 1, \dots, c, \quad j = 1, \dots, \pi, \\ F_{\text{obj}} + p & \text{otherwise,} \end{cases} \quad (23)$$

where p is the penalty term whose value is positive. In case of infeasible solutions (i.e., $n_{i\pi} < 0$), Gibbs free energy function of phase π cannot be determined due to the logarithmic terms of the activity or fugacity coefficients. So, the penalty term used for handling infeasible solutions in rPEC is given by [2]

$$p = 10 \times \sum_{i=1}^{n_{\text{unf}}} |n_{i\pi}| \quad (24)$$

where $n_{i\pi}$ is obtained from Eq. (22) and n_{unf} is the number of infeasible mole numbers (i.e., $n_{i\pi} < 0$ where $i = 1, \dots, c - r$). In this study, the resulting constrained Gibbs free energy minimization for a reactive system is solved using UBBPSO, IDE.N and IDE algorithms. The details of the rPEC problems are shown in Table 2.

5. Implementation of the methods

In this study, all the optimization algorithms and thermodynamic models are coded in Matlab. The parameters used for the algorithms are fixed for all problems tested in order to compare the

Table 2
Details of rPEC (chemical equilibrium) problems studied.

rPEC No.	System	Feed conditions	Thermodynamic models	Global optimum	Ref.
1	A1 + A2 \leftrightarrow A3 + A4 (1) Ethanol (2) Acetic acid (3) Ethyl acetate (4) Water	$n_F = (0.5, 0.5, 0.0, 0.0)$ at 355 K and 101.325 kPa	NRTL model and ideal gas. $K_{\text{eq}} = 18.670951$	−1.298000	[2,6,25]
2	A1 + A2 \leftrightarrow A3, and A4 as an inert component (1) Isobutene (2) Methanol (3) Methyl ter-butyl ether (4) n-Butane	$n_F = (0.3, 0.3, 0.0, 0.4)$ at 373.15 K and 101.325 kPa	Wilson model and ideal gas. $\Delta G^0_{\text{rxs}}/R = -4205.05 + 10.0982T - 0.26677 \ln T$ $\ln K_{\text{eq}} = -\Delta G^0_{\text{rxs}}/R$ where T is in K	−1.434267	[2]
3	A1 + A2 + 2A3 \leftrightarrow 2A4 (1) 2-Methyl-1-butene (2) 2-Methyl-2-butene (3) Methanol (4) Tert-amyl methyl ether	$n_F = (0.354, 0.183, 0.463, 0.0)$ at 355 K and 151.95 kPa	Wilson model and ideal gas. $K_{\text{eq}} = 1.057 \times 10^{-04} e^{4273.5/T}$ where T is in K	−1.226367	[2,21]
4	A1 + A2 \leftrightarrow A3 + A4 (1) Acetic acid (2) n-Butanol (3) Water (4) n-Butyl acetate	$n_F = (0.3, 0.4, 0.3, 0.0)$ at 298.15 K and 101.325 kPa	UNIQUAC model and ideal gas. $\ln K_{\text{eq}} = 450/T + 0.8$	−0.301730	[2,9]
5	A1 + A2 \leftrightarrow A3	$n_F = (0.6, 0.4, 0.0)$	Margules solution model. $g^E/R_g T = 3.6x_1x_2 + 2.4x_1x_3 + 2.3x_2x_3$ $K_{\text{eq}} = 0.9825$	−1.798377	[21]
6	A1 + A2 + 2A3 \leftrightarrow 2A4 with A5 as inert component (1) 2-Methyl-1-butene (2) 2-Methyl-2-butene (3) Methanol (4) Tert-amyl methyl ether (5) n-Pentane	$n_F = (0.1, 0.15, 0.7, 0.0, 0.05)$ at 335 K and 151.9875 kPa	Wilson model and ideal gas. $K_{\text{eq}} = 1.057 \times 10^{-04} e^{4273.5/T}$ where T is in K	−0.144508	[21]
7	A1 + A2 \leftrightarrow A3	$n_F = (0.52, 0.48, 0.0)$ at 323.15 K and 101.325 kPa	Margules solution model. $K_{\text{eq}} = 3.5$	−1.043199	[2,9]
8	A1 + A2 \leftrightarrow A3 + A4	$n_F = (0.048, 0.5, 0.452, 0.0)$ at 360 K and 101.325 kPa	NRTL model. $K_{\text{eq}} = 4.0$	−1.347857	[2,9]

robustness of the algorithm. Further, NP = 10 D for the three methods: UBBPSO, IDE and IDE.N, and TL = 50 and TR = 0.001 D are used in IDE. Altogether, there are 24 problems consisting of 8 PEC, 8 PS and 8 rPEC problems, whose details can be found in Tables 1 and 2. All these problems are multimodal with number of decision variables ranging from 2 to 36. Each problem is solved 100 times independently with a different random number seed for robust performance analysis. The performances of stochastic algorithms are compared based on success rate (SR) and average number of function evaluations (for both global and local searches) in the 100 runs (NFE), for two stopping criteria: SC-1 based on the maximum number of iterations and SC-2 based on the maximum number of iterations without improvement in the *gbest* objective function value (SC_{max}). SC-2 is also known as *ImpBest*, an improvement-based stopping criterion.

Note that NFE is a good indicator of computational efficiency since function evaluation involves extensive computations in application problems. Further, it is independent of the computer and software platform used, and so it is useful for comparison by researchers. SR is the number of times the algorithm located the global optimum to the specified accuracy, out of 100 runs. A run/trial is considered successful if the *gbest* objective function value obtained after the local optimization is within $1.0E-5$ from the known global optimum. Also, global success rate (GSR) of different algorithms is reported for all the problems. It is defined as

$$GSR = \frac{\sum_{i=1}^{np} SR_i}{np} \quad (25)$$

where *np* is the number of problems and SR_i is the individual success rate for each problem.

In total, three algorithms were tried for solving these problems; they are IDE, IDE.N and UBBPSO. Later, results of these algorithms are compared and discussed. At the end of each run by each stochastic algorithm, a local optimizer is used to continue the search to find the global optimum precisely and efficiently. This is also done at the end of different iteration levels for analysis; however, global search in the subsequent iterations is not affected by this. Since all algorithms are implemented in Matlab, sequential quadratic program (SQP) is chosen as the local optimizer. The best solution at the end of the stochastic algorithm is used as the initial guess for SQP, which is likely to locate the global optimum if the initial guess is in the global optimum region. All computations were performed on Dell Optiplex GX620 with a Pentium IV, 3.6 GHz and 2 GB of RAM, which can complete 297 MFlops (million floating-point operations) for the LINPACK benchmark program for a matrix of order 500.

6. Results and discussion

6.1. Performance of algorithms on PEC problems

First, GSR values for all PEC problems by IDE, IDE.N and UBBPSO algorithms with NP of 10 D using SC-1 are illustrated in Fig. 3. The results are collected at different iteration levels, starting from 50 to 1500 iteration level, after local optimization at each of these iteration levels. As expected, GSR improves with increasing number of iterations (Fig. 3), particularly at lower iteration levels. After 250 iterations, GSR does not improve significantly; this suggests that subsequent iterations without improvement in the results are waste of computational resources. For example, GSR of UBBPSO is 83.5% at 50 iterations; it increases to 88% at 250 iterations and 89.6% at 1500 iterations. GSR of IDE is 83%, 99.9% and 100% at 50, 250 and 1500 iterations, respectively. Results in Fig. 3 show that IDE has higher reliability and faster convergence rate compared to IDE.N and UBBPSO, for PEC problems. Further, IDE and IDE.N can achieve 100% GSR at 1500 iteration level. Thus, it is essential

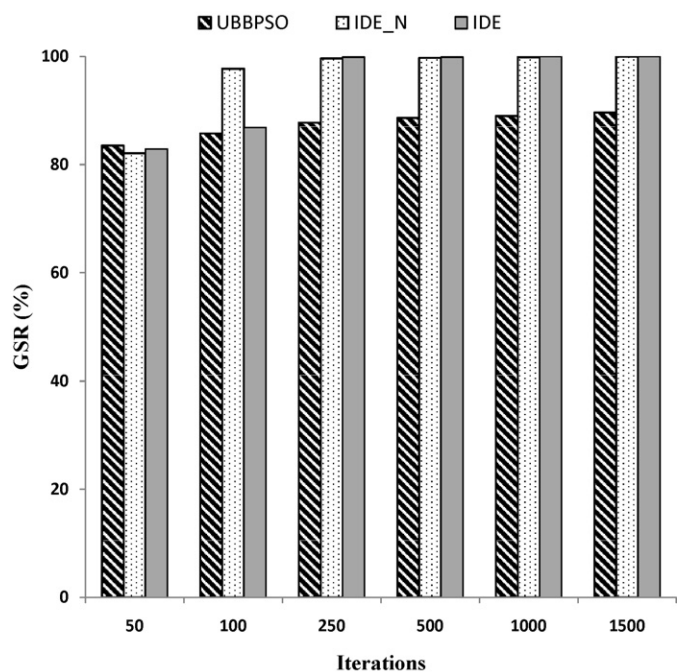


Fig. 3. Global success rate (GSR) versus iterations for PEC problems using UBBPSO, IDE.N and IDE with SC-1.

for the optimization algorithm to stop at the right time incurring least computational resources without compromising reliability of finding the global optimum.

The effect of stopping criterion, SC-2 on IDE, IDE.N and UBBPSO algorithms has also been studied on PEC problems. Table 3 summarizes SR and NFE obtained by these algorithms with $SC_{max} = 10, 25$ and 50 along with the maximum allowable iterations of 1500 (to avoid indefinite looping), all using NP = 10 D. It shows that relia-

Table 3

Success rate (SR) and number of function evaluations (NFE) of UBBPSO, IDE.N and IDE for PEC problems using SC_{max} with NP of 10 D.

PEC No.	SC_{max}	UBBPSO		IDE.N		IDE	
		SR	NFE	SR	NFE	SR	NFE
PEC-1	10	71	1303	72	955	75	886
	25	89	4151	86	1980	93	1826
	50	100	6757	92	4405	98	3407
PEC-2	10	99	1694	100	1326	100	1307
	25	100	7898	100	4803	100	3386
	50	100	10814	100	7988	100	6021
PEC-3	10	100	1816	100	1253	100	954
	25	100	12019	100	5304	100	2916
	50	100	18224	100	7396	100	4488
PEC-4	10	67	1414	85	507	83	491
	25	88	3491	95	1922	98	1371
	50	94	10121	98	4680	100	2669
PEC-5	10	8	9851	5	4101	31	1879
	25	9	16619	54	12406	100	6991
	50	10	25062	97	14599	100	13116
PEC-6	10	99	60812	99	9441	98	6318
	25	100	73893	100	26921	100	17995
	50	100	84375	100	31615	100	29251
PEC-7	10	100	22678	99	15044	100	13699
	25	100	32835	99	30052	100	35353
	50	100	37831	100	47858	100	54626
PEC-8	10	98	116054	96	17456	80	7284
	25	99	125860	100	34345	94	19463
	50	99	138266	100	38509	100	35456

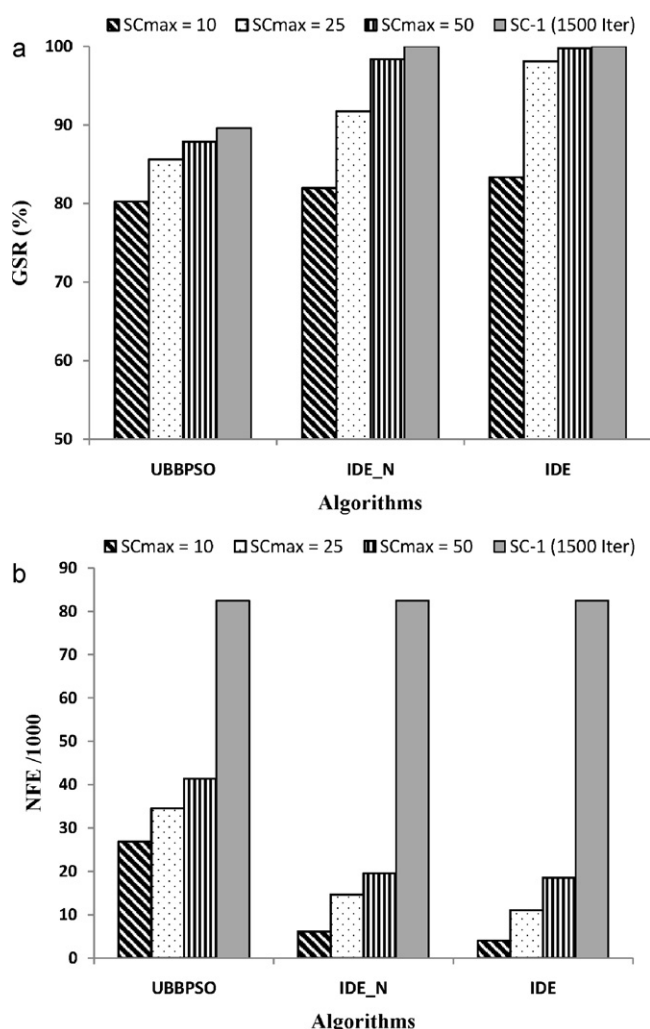


Fig. 4. Global success rate, GSR (plot a) and NFE (plot b) of UBBPSO, IDE_N and IDE for PEC problems using SC-2 ($SC_{max} = 10$, $SC_{max} = 25$ and $SC_{max} = 50$) and SC-1 (1500 iterations).

bility of the algorithms increases with SC_{max} , which requires more NFE. This is because probability to locate the global optimum region increases as the algorithms are allowed to run for more iteration. For PEC problems 1–3 and 6–8, the three algorithms obtained similar high reliability; and, for PEC problems 4 and 5, IDE and IDE_N obtained better reliability than UBBPSO. Even through the same stopping criterion is used for all the algorithms; NFE required by IDE is much less than that of IDE_N and UBBPSO. As shown in Table 3, the total NFE required by IDE for all tests on PEC problems is 271153 compared to 324866 and 823838 for IDE_N and UBBPSO, respectively. This clearly shows that IDE has faster convergence compared to the others, probably because of prevention of re-visiting the same place by tabu list and checking.

Fig. 4 summarizes GSR and NFE of IDE, IDE_N and UBBPSO algorithms with four stopping criteria. We obtain the same conclusion of higher reliability and increased NFE with higher SC_{max} . It can be observed in Fig. 4a that the use of SC-2 gives lower GSR compared to SC-1. However, with the use of $SC_{max} = 50$, the reliability of the algorithm is only slightly lower than that with SC-1 (1500 iterations). Comparison of the three algorithms shows that IDE uses least NFE to terminate the global search by SC-2. In general, SC-2 requires significantly fewer NFE compared to SC-1. Especially, with $SC_{max} = 50$, SR obtained by the algorithms is comparable to that obtained with SC-1 but uses much fewer NFE (Fig. 4). Compared to IDE_N, IDE has

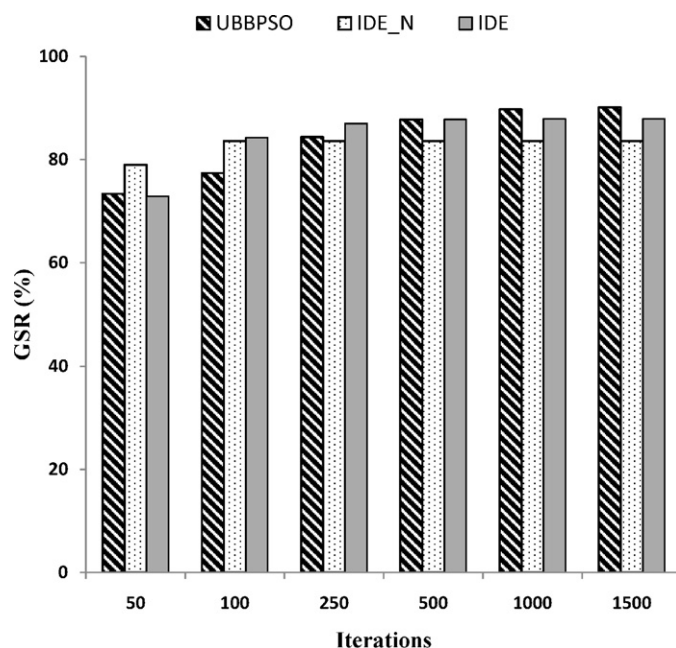


Fig. 5. Global success rate (GSR) versus iterations for PS problems using UBBPSO, IDE_N and IDE with SC-1.

achieved better reliability with fewer NFE, probably due to the tabu list and tabu check operations in IDE, which prevents revisiting the searched areas thus enhancing the global search ability. In summary, it can be concluded that IDE is better and that $SC_{max} = 50$ is a good stopping criterion for PEC problems in order to achieve both high reliability and efficiency.

6.2. Performance of algorithms on PS problems

On PS problems, similar tests using the three stochastic algorithms are performed. As expected, GSR of UBBPSO, IDE_N and IDE for all PS problems using SC-1 improves with increasing number of iterations (Fig. 5). The highest GSR is 90.1% obtained by UBBPSO algorithm. Results in Fig. 5 show that the selected PS problems are more difficult to optimize compared to PEC problems. At 50 iterations, IDE_N obtained best GSR, but from 100 to 1500 iterations, GSR of IDE_N did not improve but UBBPSO and IDE obtained better GSR. In fact, IDE_N has faster convergence but it is easily trapped at a local optimum. GSR of UBBPSO is 73% at 50 iterations; it increases to 84% at 250 iterations and 90% at 1500 iterations. On the other hand, GSR of IDE is 73%, 87% and 88% at 50, 250 and 1500 iterations, respectively. This performance indicates that reliability of the algorithm did not improve significantly after 250 iterations. Thus, it is necessary to use a suitable stopping criterion for the optimization algorithm to stop at the right time incurring least computational resources without compromising reliability of finding the global optimum.

Results on the effect of stopping criterion, SC-2 with $SC_{max} = 10$, 25 and 50 on IDE, IDE_N and UBBPSO for all PS problems are presented in Table 4. They show that reliability of the algorithm and NFE increase with increasing SC_{max} . For PS problems 1 and 5, IDE_N obtained the best reliability followed by IDE algorithm. For PS problems 2, 3, 4 and 7, the three algorithms obtained 100% SR. For PS problems 6 and 8, UBBPSO obtained the best reliability followed by IDE. However, UBBPSO and IDE_N require significantly more NFE than IDE. Even through the same stopping criterion is used for the three algorithms, NFE required by IDE is much less than that of IDE_N and UBBPSO. For example, for PS-8 with $SC_{max} = 50$, UBBPSO needs 147533 NFE but IDE requires only 70996 NFE. As shown in

Table 4

Success rate (SR) and number of function evaluations of UBBPSO, IDE.N and IDE for PS problems using SC_{max} with NP of 10 D.

PS No.	SC_{max}	UBBPSO		IDE.N		IDE	
		SR	NFE	SR	NFE	SR	NFE
PS-1	10	85	1274	93	594	89	582
	25	97	3686	100	2518	99	2631
	50	99	6051	100	3553	100	3789
PS-2	10	100	2536	100	2227	100	1448
	25	100	7880	100	5756	100	6536
	50	100	10515	100	6571	100	9053
PS-3	10	100	3342	100	1741	100	1250
	25	100	8906	100	4721	100	4747
	50	100	14856	100	6144	100	7619
PS-4	10	100	827	100	822	100	781
	25	100	2310	100	2339	100	2437
	50	100	4096	100	3369	100	3321
PS-5	10	56	21272	69	4033	46	2051
	25	78	31777	99	10573	82	10055
	50	92	44606	100	13285	99	19487
PS-6	10	73	74920	47	9296	45	3633
	25	90	93490	62	25816	72	15148
	50	94	105846	62	30045	76	38145
PS-7	10	100	64859	100	18020	100	5417
	25	100	90780	100	45681	100	10253
	50	100	105697	100	50074	100	14161
PS-8	10	25	127894	7	17033	9	5470
	25	26	140833	7	31647	10	20997
	50	26	147533	7	36894	21	70996

Table 4, the total NFE required for all PS problems by IDE is 260007 compared to 332752 and 1115786 required by IDE.N and UBBPSO, respectively. Thus, UBBPSO requires more than 4 times the total NFE required by IDE. Hence, IDE has faster convergence compared to IDE.N and UBBPSO, probably due to tabu list and checking.

Fig. 6 summarizes GSR and NFE of IDE, IDE.N and UBBPSO with four stopping criteria for PS problems. We obtain the same conclusion of higher reliability and increased NFE with higher SC_{max} . It can be seen from Fig. 6a that the use of SC-2 gives similar GSR compared to SC-1; especially, GSR of IDE.N with different stopping criteria is similar. Among the three algorithms, IDE uses least NFE to terminate the global search in case of $SC_{max} = 10$ and $SC_{max} = 25$ whereas IDE.N uses the least NFE to terminate the global search progress for $SC_{max} = 50$ (Fig. 6b). In general, use of SC-2 has significantly reduced NFE compared to that using SC-1. As for PEC problems, it can be concluded that IDE is better and that $SC_{max} = 50$ is a good stopping criterion for PS problems in order to achieve both high reliability and efficiency.

Low SR is obtained for PS-5, 6 and 8 compared to other PS problems (Table 4). This is probably because of using the same NP = 10 D for comparing the three algorithms studied in this paper. In general, NP is a user-specified parameter; it does not need to be fine-tuned and just a few typical values can be tried according to the pre-estimated complexity of the given problem. So PS-5, 6 and 8 are solved by IDE with higher population size of NP = 50 D. The results show that IDE can obtain 100% SR using 24930 NFE for PS-5, 100% SR using 61820 NFE for PS-6 and 76% SR using 384416 NFE for PS-8. It is clear that reliability of the algorithm increases with population size but this requires more computational effort. This is reasonable because larger population size enables more thorough exploration of search space. In general, stochastic optimization methods provide only a probabilistic guarantee of locating the global optimum, and their proofs for numerical convergence usually state that the global optimum will be identified in infinite time with probability 1

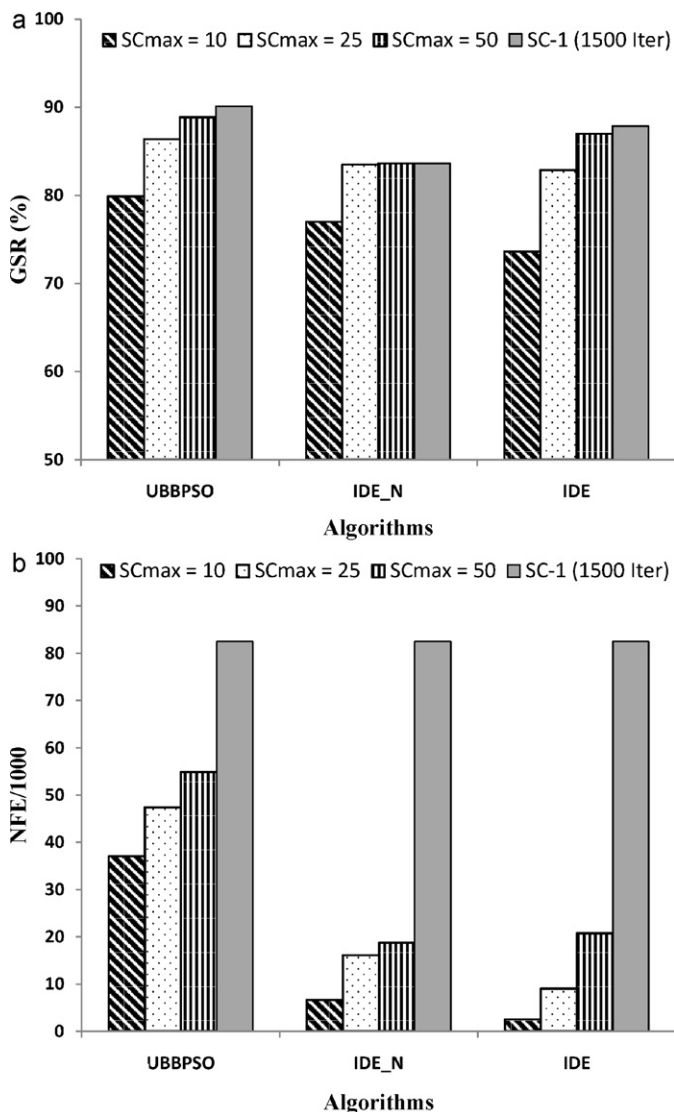


Fig. 6. Global success rate, GSR (plot a) and NFE (plot b) of UBBPSO, IDE.N and IDE for PS problems using SC-2 ($SC_{max} = 10$, $SC_{max} = 25$ and $SC_{max} = 50$) and SC-1 (1500 iterations).

[42–44]. So, better performance of stochastic methods is expected if more iterations and/or larger population size are used.

To analyze further, performance if IDE is compared with that of other stochastic optimization algorithms, namely, SA, very fast SA (VFSA), modified version of direct search SA (MDSA) and stochastic differential equations algorithm (SDE) for PS-3, 5, 6 and 8 reported in Bonilla-Petriciolet et al. [21], in Table 5. From this table, it is clear that IDE uses about 10 times fewer NFE compared to the other four stochastic methods for solving PS-3, 5 and 6 with the same reliability. SR of IDE is lower for PS-8 but using significantly less NFE, and this may be due to the different stopping criteria and population size used in the algorithms. Note that some methods may give better reliability using the stopping criteria based on either known global optimum or the number of generations [21]. However, the use of known global optimum is not applicable for new problems whose global optima are unknown, and use of number of generations may require large computational time.

Bonilla-Petriciolet et al. [6] compared various PSO algorithms for solving the PEC and PS problems. Their results suggest that the classical PSO outperforms other variants of PSO. So, the performance of IDE and UBBPSO is compared with two classical PSO algorithms

Table 5
Comparison of SR and NFE of IDE with other stochastic algorithms for selected PS problems.

PS No.		MDSA [21]	SA [21]	VFSA [21]	SDE [21]	IDE (NP = 10 D)	IDE (NP = 50 D)
PS-3	NFE	82611	92422	42944	449574	1250	–
	SR	100	100	100	100	100	–
PS-5	NFE	263980	266926	129534	565142	19487	24930
	SR	100	100	96	100	99	100
PS-6	NFE	752571	700865	362097	476029	38145	61820
	SR	100	100	100	100	76	100
PS-8	NFE	1167211	1104901	581396	958515	70996	384416
	SR	100	100	98	98	21	76

reported in Bonilla-Petriciolet et al. [6] for both PEC and PS problems, with different types of stopping criteria in Fig. 7. The two PSO algorithms are classical PSO with quasi-Newton method (PSO-CQN) and classical PSO with Nelder-Mead simplex method (PSO-CNM). Fig. 7a shows that IDE achieved the best reliability at 100 iterations or more, compared to UBBPSO, PSO-CQN and PSO-CNM. UBBPSO achieved the second best reliability at different iteration levels among the four algorithms tested with SC-1. The reliability comparison of the four algorithms with SC-2 stopping criterion is shown in

Fig. 7b. IDE gave the highest GSR even with SC-2 as stopping criteria with $SC_{\max} = 25$ and 50, among the four algorithms. With stopping criterion, $SC_{\max} = 10$, UBBPSO obtained slightly better GSR than IDE but much better GSR than PSO-CQN and PSO-CNM. Overall, IDE is superior to UBBPSO, PSO-CQN and PSO-CNM algorithms for PEC and PS problems.

6.3. Performance of algorithms on rPEC problems

GSR of UBBPSO, IDE.N and IDE algorithms for all rPEC problems using SC-1 is illustrated in Fig. 8. It can be seen that GSR generally improves with increasing number of iterations for these problems as well. The highest GSR is 91% obtained by IDE. At 50 iterations, IDE.N obtained best GSR, but from 250 to 1500 iterations, its GSR did not improve; on the other hand, UBBPSO and IDE obtained better GSR at higher iterations. GSR of UBBPSO is 80% at 50 iterations, and it increases to 87% at 250 iterations and 90% at 1500 iterations. GSR of IDE is 77%, 85% and 91% at 50, 250 and 1500 iterations, respectively. In short, UBBPSO, IDE.N and IDE algorithms obtained good GSR of 80% or more (Fig. 8).

Results obtained on the effect of stopping criteria on the three algorithms using SC-2 with $SC_{\max} = 6D$, 12D and 24D, for rPEC problems are summarized in Table 6. Note that SC_{\max} values used for each rPEC problem were those used by Bonilla-Petriciolet et al. [2] so that the present results can be compared with those in [2].

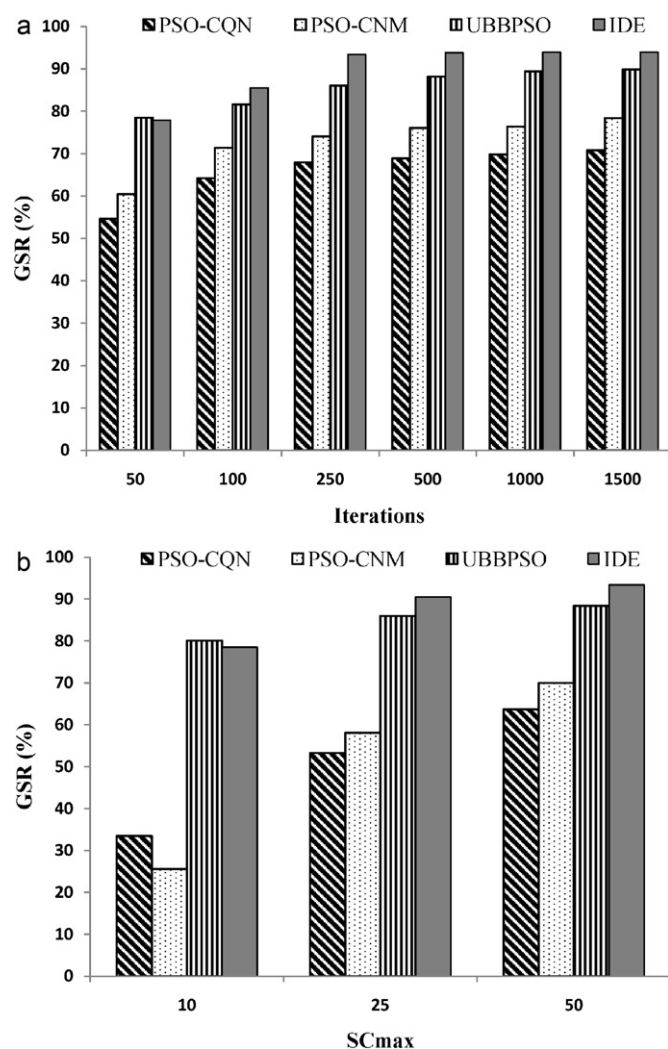


Fig. 7. Global success rate of PSO-CQN, PSO-CNM and IDE in PEC and PS problems using: (a) SC-1 and (b) SC-2 ($SC_{\max} = 10$, $SC_{\max} = 25$ and $SC_{\max} = 50$) as stopping criteria.

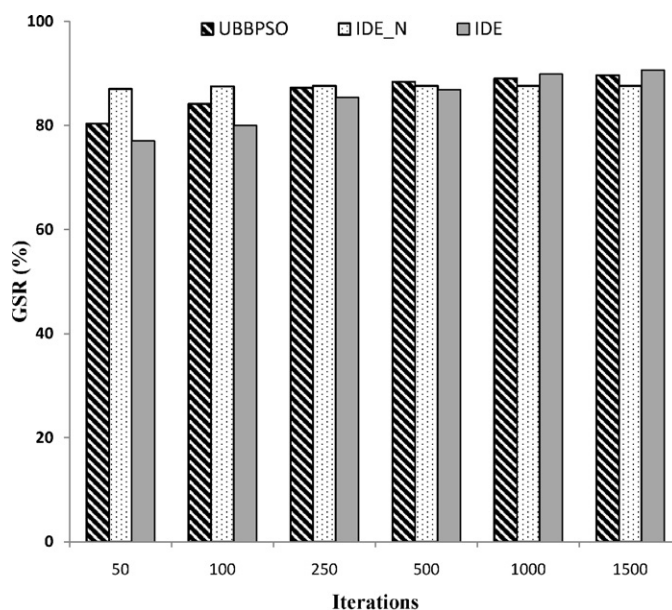


Fig. 8. Global success rate (GSR) versus iterations for rPEC problems using UBBPSO, IDE.N and IDE with SC-1.

Table 6

Success rate (SR) and number of function evaluations of UBBPSO, IDE.N and IDE for rPEC problems using SC_{max} with NP of 10 D.

rPEC No.	SC_{max}	UBBPSO		IDE.N		IDE	
		SR	NFE	SR	NFE	SR	NFE
rPEC-1	6 D	100	21585	100	14484	100	5585
	12 D	100	47720	100	16102	100	14720
	24 D	100	68953	100	19387	100	16641
rPEC-2	6 D	100	33794	100	13012	100	12698
	12 D	100	61252	100	14511	100	16559
	24 D	100	70037	100	17511	100	18885
rPEC-3	6 D	100	14896	100	14928	100	8143
	12 D	100	29376	100	16461	100	14157
	24 D	100	47986	100	19460	100	27034
rPEC-4	6 D	10	23107	1	11676	9	10312
	12 D	13	34655	1	14394	17	13765
	24 D	14	44068	1	19014	17	20763
rPEC-5	6 D	93	14792	98	8794	90	2644
	12 D	98	32370	99	12160	98	5664
	24 D	98	43177	100	15793	100	11264
rPEC-6	6 D	100	39571	100	18361	100	19504
	12 D	100	43742	100	22175	100	21135
	24 D	100	50421	100	28628	100	29150
rPEC-7	6 D	90	16129	99	9229	90	2546
	12 D	97	28041	100	11945	97	4680
	24 D	98	33326	100	14001	98	10330
rPEC-8	6 D	100	22954	100	15487	100	5780
	12 D	100	25941	100	18652	100	10411
	24 D	100	31014	100	23720	100	23117

Table 6 shows that reliability of the algorithm increases with SC_{max} but requires more NFE. For rPEC-1, 2, 3, 6 and 8, UBBPSO, IDE.N and IDE algorithms obtained 100% SR with different SC_{max} tried. For rPEC problem 4, IDE obtained the best reliability followed by UBBPSO and IDE.N, but its SR is only 17%. So rPEC-4 is solved by IDE using larger population of NP = 50 D. The results show that IDE can obtain 88% SR at 118108 NFE. Among the three algorithms, NFE required by IDE is much less than that by IDE.N and UBBPSO. As shown in Table 5, total NFE required by IDE for all rPEC problems is 325487 compared to 389885 by IDE.N and 878907 by UBBPSO. On the other hand, mean SR for the three algorithms is almost the same. Thus, our results show that IDE has faster convergence compared to IDE.N and UBBPSO for the same reliability.

Fig. 9 shows the performance of GSR and NFE of IDE, IDE.N and UBBPSO with different stopping criteria for rPEC problems. Again, we conclude that the higher the SC_{max} , the better the reliability of the algorithm is, and that the use of SC-2 gives similar GSR compared to SC-1 (Fig. 9a). However, IDE.N gave almost the same GSR with different stopping criteria. Among IDE, IDE.N and UBBPSO, IDE uses least NFE to terminate the global search progress via SC-2. Between IDE and IDE.N, results show that IDE provides better reliability with fewer NFE (Fig. 9), probably due to enhancement of global search ability by tabu list and tabu check operations in IDE. For optimal efficiency and reliability of IDE, $SC_{max} = 12$ D is a good stopping criterion for rPEC problems.

Recently, SA, DETL and GA have been tested for rPEC problems in [2]. All these stochastic algorithms were run 100 times independently in FORTRAN environment. At the end of every run, a deterministic local optimizer (namely, DBCONF of IMSL library) was activated. The performance of IDE is compared with that GA, SA and DETL for rPEC problems, in Table 7. The data of IDE in Table 7 is slightly different from those in Table 5 because NFE given in [2] is based on successful runs only (and not all runs). For fair comparison, similar data of IDE are given in Table 7. From this table, it is clear that IDE achieved the best reliability compared to SA, GA and

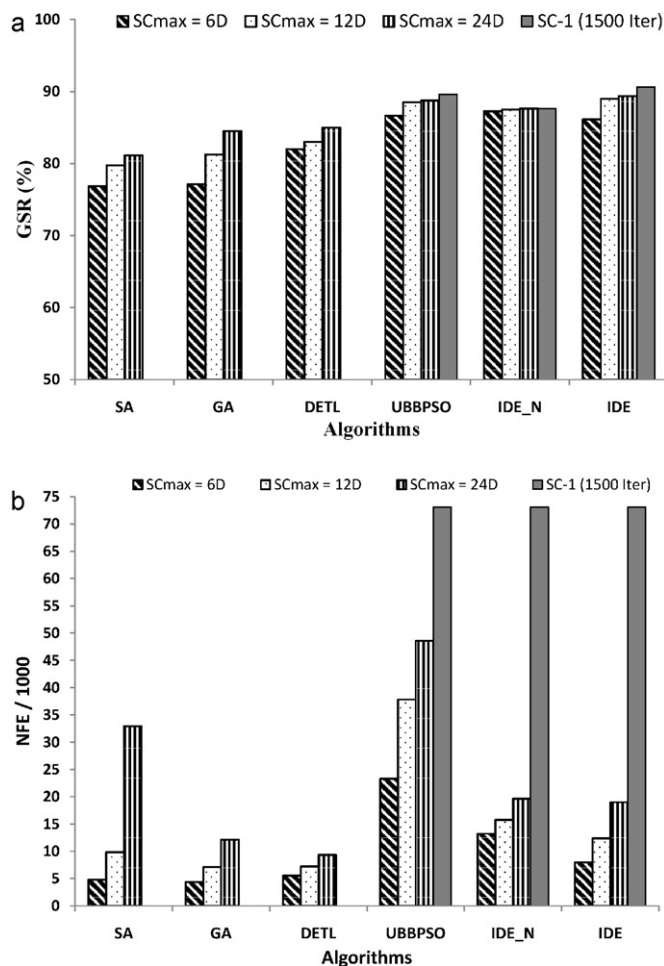


Fig. 9. Global success rate, GSR (plot a) and NFE (plot b) of UBBPSO, IDE.N, IDE, SA, GA and DETL for rPEC problems using SC-2 ($SC_{max} = 6$ D, $SC_{max} = 12$ D and $SC_{max} = 24$ D) and SC-1 (1500 iterations).

DETL at $SC_{max} = 6$ D, 12 D and 24 D. For example, for rPEC problems 5 and 7, IDE obtained much better SR than the other algorithms. However, IDE requires more NFE compared to GA and DETL. This indicates that GA and DETL have faster convergence rate for rPEC problems but they can be trapped at the local optimum in several runs. The computational time (in s) for solving rPEC problems using IDE algorithm with $SC_{max} = 6$ D, 12 D and 24 D is reported in the last column of Table 7. Obviously, CPU times increases with NFE. Further, there is nearly linear relationship between NFE and CPU time, which supports the use of NFE for comparison.

For rPEC 4, at $SC_{max} = 6$ D, GA and SA are better than IDE in terms of both SR and NFE. When $SC_{max} = 12$ D, IDE is more reliable than GA and SA but with more NFE. When $SC_{max} = 24$ D, the IDE is better than GA and SA in both SR and NFE. This is probably due to the self-adaptive strategy of IDE which needs more generations initially for adaptive tuning of algorithm parameters. For high reliability of GSR = 85%, GA requires SC_{max} of 24 D and NFE of 12138 whereas IDE requires SC_{max} of 12 D and NFE of 12375 to obtain 89% GSR. This shows that IDE uses 2% more NFE to improve GSR by 4%. Further, IDE has fewer parameters to be tuned, which makes the algorithm more robust. In summary, the present results indicate that IDE offers a good balance between diversification and intensification stages for reliable and efficient phase equilibrium calculations in both reactive and non-reactive systems. Compared to other stochastic methods, its reliability and efficiency are generally better for solving these thermodynamic problems.

Table 7Success rate (SR) and number of function evaluations of SA, GA, DETL and IDE for rPEC problems using SC_{max} with NP of 10 D.

rPEC No.	SC _{max}	SA [2]		GA [2]		DETL [2]		IDE		Time (s)
		SR	NFE	SR	NFE	SR	NFE	SR	NFE	
rPEC-1	6 D	93	5544	89	4650	100	7791	100	5585	3
	12 D	99	10818	98	7866	100	9787	100	14720	7
	24 D	100	44083	99	13228	100	11548	100	16641	9
rPEC-2	6 D	98	4703	99	4465	100	4237	100	12698	6
	12 D	100	10159	100	7363	100	5708	100	16559	10
	24 D	100	35837	100	12464	100	6665	100	18885	12
rPEC-3	6 D	95	4740	98	4670	100	6366	100	8143	5
	12 D	100	9299	100	7484	100	8017	100	14157	8
	24 D	100	32630	100	12680	100	10893	100	27034	15
rPEC-4	6 D	11	4460	15	4234	1	10621	9	6799	3
	12 D	10	8396	11	7065	1	15420	17	11647	6
	24 D	11	29589	17	12136	2	22043	17	11652	6
rPEC-5	6 D	71	2401	63	2403	88	2235	90	2706	2
	12 D	76	4725	73	4224	86	2705	98	5685	4
	24 D	79	9508	70	8292	92	3205	100	11264	7
rPEC-6	6 D	99	8038	100	6420	100	4859	100	19504	11
	12 D	100	18759	100	10708	100	5994	100	21135	14
	24 D	100	66336	100	17623	100	8532	100	29150	19
rPEC-7	6 D	56	2690	58	2913	65	3955	90	2573	1
	12 D	55	5389	70	4584	77	4985	97	4682	2
	24 D	61	13599	90	8257	85	5713	98	10338	5
rPEC-8	6 D	92	5733	95	4906	100	4057	100	5780	2
	12 D	98	11096	98	7777	100	5036	100	10411	4
	24 D	98	31887	100	12420	100	6124	100	23117	10

7. Conclusions

The stochastic global optimization algorithms, namely, UBBPSO, IDE_N and IDE studied in this work have fewer parameters to be tuned. The performance of these algorithms has been tested and compared for solving PEC, rPEC and PS problems. IDE was found to be the overall best performer across different problems tried. Results for IDE and IDE_N confirm that use of tabu radius and tabu list improves reliability and decreases computational effort although it involves two parameters. Comparison of IDE with PSO variants for both PEC and PS problems shows that the IDE provides higher reliability and efficiency. Comparison of IDE with SA, GA and DETL suggests that the former provides higher reliability for rPEC problems. The stopping criterion, SC-1 gives slightly better reliability than SC-2 at the expense of computational resources, and the use of SC_{max} can significantly reduce the computational effort for solving PEC, rPEC and PS problems without much effect on the reliability of the stochastic algorithms studied.

References

- [1] M. Srinivas, G.P. Rangaiah, *Comput. Chem. Eng.* 31 (2007) 760–772.
- [2] A. Bonilla-Petriciolet, G.P. Rangaiah, J.G. Segovia-Hernández, *Fluid Phase Equilib.* 300 (2011) 120–134.
- [3] M. Srinivas, G.P. Rangaiah, *Ind. Eng. Chem. Res.* 46 (2007) 3410–3421.
- [4] D. Reynolds, A.J. Mulholland, J. Gomatam, *J. Math. Chem.* 22 (1997) 25–27.
- [5] G.P. Rangaiah, *Fluid Phase Equilib.* 187–188 (2001) 83–109.
- [6] A. Bonilla-Petriciolet, J.G. Segovia-Hernández, *Fluid Phase Equilib.* 289 (2010) 110–121.
- [7] C.C.R.S. Rossi, L. Cardozo-Filho, R. Guirardello, *Fluid Phase Equilib.* 278 (2009) 117–128.
- [8] F. Jalali, J.D. Seader, S. Khaleghi, *Comput. Chem. Eng.* 32 (2008) 2333–2345.
- [9] S.K. Wasylkiewicz, S. Ung, *Fluid Phase Equilib.* 175 (2000) 253–272.
- [10] F. Jalali, J.D. Seader, *Comput. Chem. Eng.* 23 (1999) 1319–1331.
- [11] G.I. Burgos-Solórzano, J.F. Brennecke, M.A. Stadtherr, *Fluid Phase Equilib.* 219 (2004) 245–255.
- [12] A.C. Sun, W.D. Seider, *Fluid Phase Equilib.* 103 (1995) 213–249.
- [13] C.C.R.S. Rossi, M.E. Berezuk, L. Cardozo-Filho, R. Guirardello, *Comput. Chem. Eng.* 35 (2011) 1226–1237.
- [14] S.T. Harding, C.A. Floudas, *AIChE J.* 46 (2000) 1422–1440.
- [15] A. Cheung, C.S. Adjiman, P. Kolar, T. Ishikawa, *Fluid Phase Equilib.* 194 (2002) 169–183.
- [16] W.A. Wakeham, R.P. Stateva, *Rev. Chem. Eng.* 20 (2004) 1–56.
- [17] D.V. Nichita, S. Gomez, E. Luna, *Comput. Chem. Eng.* 26 (2002) 1703–1724.
- [18] D.V. Nichita, S. Gomez, E. Luna, *Fluid Phase Equilib.* 194 (2002) 411–437.
- [19] D.V. Nichita, F. Garcia-Sanchez, S. Gomez, *Chem. Eng. J.* 140 (2008) 509–520.
- [20] D.V. Nichita, S. Gomez, E. Luna, *J. Can. Pet. Technol.* 43 (2004) 13–16.
- [21] A. Bonilla-Petriciolet, R. Vázquez-Román, G.A. Iglesias-Silva, K.R. Hall, *Ind. Eng. Chem. Res.* 45 (2006) 4764–4772.
- [22] M. Srinivas, G.P. Rangaiah, *Comput. Chem. Eng.* 30 (2006) 1400–1415.
- [23] M. Srinivas, G.P. Rangaiah, in: G.P. Rangaiah (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, 2010, pp. 465–504.
- [24] Y.S. Teh, G.P. Rangaiah, *Trans. IChemE* 80 (2002) 745–759.
- [25] Y.S. Teh, G.P. Rangaiah, *Comput. Chem. Eng.* 27 (2003) 1665–1679.
- [26] J. Kennedy, R. Eberhart, *Proceedings of the Sixth International Symposium on Micromachine Human Science*, 1995, pp. 39–43.
- [27] Y. Shi, R.C. Eberhart, *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1998.
- [28] J. Kennedy, *Proceedings of the IEEE on Swarm Intelligence Symposium*, 2003.
- [29] H. Zhang, D.D. Kennedy, G.P. Rangaiah, A. Bonilla-Petriciolet, *Fluid Phase Equilib.* 301 (2011) 33–45.
- [30] R. Storn, K. Price, *J. Global Optim.* 11 (1997) 341–359.
- [31] B.V. Babu, S.A. Munawar, *Chem. Eng. Sci.* 62 (2007) 3720–3739.
- [32] R. Angira, A. Santosh, *Comput. Chem. Eng.* 31 (2007) 1055–1063.
- [33] K.M. Yerramsetty, C.V.S. Murty, *Comput. Chem. Eng.* 32 (2008) 1861–1876.
- [34] O.S. Soliman, L.T. Bui, *IEEE Congress on Evolutionary Computation*, 2008, pp. 2837–2842.
- [35] M.G.H. Omran, A.P. Engelbrecht, A. Salman, *Eur. J. Oper. Res.* 196 (2009) 128–139.
- [36] A.K. Qin, V.L. Huang, P.N. Suganthan, *IEEE Trans. Evol. Comput.* 13 (2009) 398–417.
- [37] H. Zhang, G.P. Rangaiah, *Asia-Pac. J. Chem. Eng.* 6 (2011) 379–390.
- [38] J.E. Baker, *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, 1987.
- [39] C.M. McDonald, C.A. Floudas, *Comput. Chem. Eng.* 19 (1995) 1111–1141.
- [40] A. Mitsos, P.I. Barton, *AIChE J.* 53 (2007) 2131–2147.
- [41] W.D. Seider, S. Widagdo, *Fluid Phase Equilib.* 123 (1996) 283–303.
- [42] G. Rudolph, *IEEE Trans. Neur. Network* 5 (1994) 96–101.
- [43] T. Niewierowicz, R. Escarela-Perez, E. Campero-Littlewood, *Comput. Electr. Eng.* 29 (2003) 505–522.
- [44] T. Weise (Ed.), *Global Optimization Algorithm: Theory and Application*, 2008.