Fluid Phase Equilibria 301 (2011) 33–45

Contents lists available at ScienceDirect

Fluid Phase Equilibria

journal homepage: www.elsevier.com/locate/fluid

Novel bare-bones particle swarm optimization and its performance for modeling vapor–liquid equilibrium data

Haibo Zhang[a], Devid Desfreed Kennedy[a], Gade Pandu Rangaiah[a,*], Adrián Bonilla-Petriciolet[b]

[a] Department of Chemical and Biomolecular Engineering, Engineering Drive 4, National University of Singapore, Singapore 117576, Singapore
[b] Department of Chemical Engineering, Instituto Tecnológico de Aguascalientes, México 20256, Mexico

ARTICLE INFO

Article history:
Received 28 June 2010
Received in revised form 21 October 2010
Accepted 27 October 2010
Available online 9 November 2010

Keywords:
Global optimization
Particle swarm optimization
Bare-bones PSO
Vapor–liquid equilibrium modeling

ABSTRACT

Parameter estimation in models plays a significant role in developing mathematical models which are used for understanding and analyzing physical, chemical and biological systems. Parameter estimation problems for vapor–liquid equilibrium (VLE) data modeling are very challenging due to the high non-linearity of thermodynamic models. Recently, stochastic global optimizations and their applications to these problems have attracted greater interest. Of the many stochastic global optimization methods, Bare-bones particle swarm optimization (BBPSO) is attractive since it has no parameters to be tuned by the user. In this study, modifications are introduced to the original BBPSO using mutation and crossover operators of differential evolution algorithm to update certain particles in the population. The performance of the resulting algorithm is tested on 10 benchmark functions and applied to 16 VLE problems. The performance of the proposed BBPSO for VLE modeling is compared with that of other stochastic algorithms, namely, differential evolution (DE), DE with tabu list, genetic algorithm and simulated annealing in order to identify their relative strengths for VLE data modeling. The proposed BBPSO is shown to be better than or comparable to the other stochastic global optimization algorithms tested for parameter estimation in modeling VLE data.

1. Introduction

Parameter estimation is very important in developing mathematical models of physical, chemical and biochemical processes [1]. Parameter estimation in a (thermodynamic) model refers to determining the values of model parameters that best fit the model to the given experimental data set. The mathematical formulation involves the minimization of a suitable objective function subject to constraints arising from the model equations. For most of the thermodynamic models, the objective function and constraints are often highly nonlinear and non-convex. Hence, it is necessary to estimate the model parameters using global optimization methods because traditional optimization methods can only provide a local solution [2]. Hence, application of global optimization to estimate the parameters in thermodynamic models is receiving greater interest in the recent past.

Several researchers have studied the parameter estimation in vapor–liquid-equilibrium (VLE) data modeling problems. The challenges and difficulties identified in solving these problems are convergence to a local optimum, flat objective function, badly scaled model functions, non-differential terms in thermodynamic equations and/or large number of optimization variables [2–6]. Thus, it is necessary to develop a reliable global optimization method in order to overcome these difficulties.

Many deterministic and stochastic algorithms have been proposed for finding the global optimum, particularly in the past two decades. The former methods can guarantee converge to the global optimum but they usually require certain properties such as continuity; in some cases, reformulation of the problems are needed depending on the characteristic of the thermodynamic models. In contrast, stochastic methods are quite simple to implement and use, and they do not require any assumptions or transformation of the original problems [6]. The stochastic algorithms include simulated annealing (SA), genetic algorithms (GA), tabu search, differential evolution (DE) and particle swarm optimization (PSO). Of these, PSO is one of the recent techniques and is the focus of this study. It is a population-based optimization technique proposed by Eberhart and Kennedy, and is motivated by the social behavior of animals such as bird flocking and fish schooling [7].

There have been many modifications to the original PSO algorithm to improve the efficiency and robustness of the search. Parsopoulos and Vrahatis [8] proposed unified PSO where the particle update is influenced by both local best (*lbest*) and global best (*gbest*). Learning strategy based on historical best information of all particles has been employed in comprehensive learning PSO [9]. Subsequently, Clerc and Kennedy [10] proposed a constriction

* Corresponding author. Tel.: +65 874 2187; fax: +65 779 1936.
E-mail address: chegpr@nus.edu.sg (G.P. Rangaiah).

0378-3812/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.fluid.2010.10.025

Nomenclature

BB	original bare-bones algorithm
BBExp	alternative original bare-bones algorithm
BBPSO	bare-bones PSO
BBPSO-MC	BBPSO with mutation and crossover operations
c	number of components for VLE problems
c_1 and c_2	acceleration coefficients of the particle
D	dimension of (number of variables in) the optimization problem
DE	differential evolution
DETL	differential evolution with tabu list
EIV	error-in-variable
$F(X)$ or F_{obj}	objective function for VLE problem
f_i	model for dependent variable i
GA	genetic algorithm
GSR	global success rate
g	vector of model functions
$gbest^k = (gbest^{1,k}, \ldots, gbest^{D,k})$	position of the global best particle at kth iteration
I	initialization range for benchmark function
I_KM	another set of initialization range for benchmark function
i	particle counter from 1 to NP
K	constriction factor
k	iteration counter from 1 to it_max
LS	least squares
$lbest_i^k = (lbest_i^{1,k}, \ldots, lbest_i^{D,k})$	position of the ith local best particle at kth iteration
m	number of dependent variables for VLE problem
NFE	number of function evaluations
NP	number of particles or population size
NRTL	non-random-two-liquid
$N()$	Gaussian normal distribution
$ndat$	number of experiments for VLE problem
$nest$	number of state variables for VLE problem
np	number of problems
$npar$	number of parameters for VLE models (NRTL, Wilson and UNIQUAC models)
$nbest_i^k = (nbest_i^{1,k}, \ldots, nbest_i^{D,k})$	position of the ith neighborhood best particle at kth iteration
$\boldsymbol{P}$	pressure (mmHg)
PS	PSO algorithm used in [6]
PSO	particle swarm optimization
$\boldsymbol{P}_i^0 = (P_1^0, \ldots, P_{ndat}^0)$	vapor pressure of pure component i (mmHg)
$\boldsymbol{P}^t = (P_1^t, \ldots, P_{ndat}^t)$	unknown "True" pressure (mmHg)
$pbest_1^k = (pbest_i^{1,k}, \ldots, pbest_i^{D,k})$	position of the ith previously best particle at k^{th} iteration
q_{ij}	ith dependent variable from jth experiment
$Rand1_i = (Rand\_1_i^1, \ldots, Rand\_1_i^D)$	first random number in the range [0,1]
$Rand2_i = (Rand\_2_i^1, \ldots, Rand\_2_i^D)$	second random number in the range [0,1]
$r_j = (r_{1,j}, \ldots, r_{nl,j})$	nl independent variables for VLE problem
S	search range for benchmark functions
SA	simulated annealing
SC-1	stopping criterion 1 used for VLE problems
SC-2	stopping criterion 2 used for VLE problems
SC$_{\max}$	maximum allowable consecutive iterations without improvement in $gbest$
SR	success rate
Std	standard deviation
$T = (T_1, \ldots, T_{ndat})$	temperature (°C)
UNIQUAC	UNIversal QUAsi-Chemical
$U[0,1]$	uniform distribution from zero to one
VLE	vapor–liquid equilibrium
VLE-LS	VLE based on LS approach
VLE-EIV	VLE based on EIV approach
$V_i^k = (V_i^{1,k}, \ldots, V_i^{D,k})$	velocity of the ith particle at kth iteration
$\boldsymbol{X}_{\max}$	upper bound for the position of the particle
$\boldsymbol{X}_{\min}$	lower bound for the position of the particle
$X_i^k = (X_i^{1,k}, \ldots, X_i^{D,k})$	position of the ith particle at kth iteration
$x_i^{\exp} = (x_{i,1}^{\exp}, \ldots, x_{i,ndat}^{\exp})$	experimental liquid mole fraction of component i
$y_i^{\exp} = (y_{i,1}^{\exp}, \ldots, y_{i,ndat}^{\exp})$	experimental vapor mole fraction of component i
z_{ij}	state variables
$z_i^t = (z_{i,1}^t, \ldots, z_{i,ndat}^t)$	unknown "True" value of i^{th} state variable
$\gamma_i = (\gamma_{i,1}, \ldots, \gamma_{i,ndat})$	activity coefficient of component i
$\gamma_i^{calc} = (\gamma_{i,1}^{calc}, \ldots, \gamma_{i,ndat}^{calc})$	calculated value for the activity coefficient of component i
$\gamma_i^{\exp} = (\gamma_{i,1}^{\exp}, \ldots, \gamma_{i,ndat}^{\exp})$	experimental value for the activity coefficient of component i
$\sigma_i = (\sigma_i^1, \ldots, \sigma_i^D)$	standard deviation vector of ith particle
$\theta = (\theta_1, \ldots, \theta_{npar})$	model parameters for VLE problem
λ	number in the range of [0,1]
$\mu_i = (\mu_i^1, \mu_i^2, \ldots, \mu_i^D,)$	mean vector of the ith particle's $pbest$ and $nbest$ vector

coefficient (K) to control the convergence of the particles. Recently, varying population size of the particle swarm has been introduced in efficient population utilization strategy PSO [11]. PSO algorithm has found many applications in engineering field. In chemical engineering, Bonilla-Petriciolet and Segovia-Hernández [5] tested the application of PSO algorithm for phase stability and equilibrium calculations in multi-component reactive and non-reactive systems. Bonilla-Petriciolet et al. [6] studied the application of PSO to parameter estimation in vapor–liquid equilibrium modeling problems.

A parameter-free PSO, known as bare-bones PSO, uses Gaussian normal distribution to update the particles in the population. It does not involve acceleration coefficient and velocity. Its performance has been found to be competitive, and a number of BBPSO algorithms have been proposed in recent years. Bare-bones approach has also been applied in differential evolution [12]. Krohling and Mendel [13] introduced jump approach in BBPSO algorithm to discourage premature convergence of the population. Motivated by the desirable features of BBPSO, a new BBPSO incorporating mutation and crossover operators of DE is developed in order to enhance the global search capability. The performance of the proposed method is first tested on benchmark functions and then applied to parameter estimation in VLE modeling using both least squares and error-in-variable approaches. Results obtained by the proposed BBPSO are compared with those in the literature and discussed.

The remainder of this paper is organized as follows. The classical PSO and BBPSO algorithm are described in Section 2. The development of the proposed BBPSO algorithm is presented in Section 3. Section 4 focuses on the performance of the proposed BBPSO algorithm on benchmark functions. The description of VLE data modeling problems and the performance of the proposed BBPSO algorithm are discussed in Section 5. Finally, the conclusions from this study are given in Section 6.

2. Description of PSO and Bare-Bones PSO methods

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

$$\text{Minimize}\, F(\boldsymbol{X}) \tag{1}$$

with respect to D decision variables: $\boldsymbol{X} = (X^1, X^2, \ldots, X^d, \ldots, X^D)$. The upper and lower bounds of these variables are $X^1_{\max}, X^2_{\max}, \ldots X^d_{\max}, \ldots X^D_{\max}$, and $X^1_{\min}, X^2_{\min}, \ldots X^d_{\min}, \ldots X^D_{\min}$, respectively. The original PSO algorithm [7] updates the particles in the population by:

$$V_i^{d,k+1} = V_i^{d,k} + c_1 * Rand1_i^d * (pbest_i^{d,k} - X_i^{d,k}) + c_2 * Rand2_i^d *(gbest^d - X_i^{d,k}) \tag{2a}$$

$$X_i^{d,k+1} = X_i^{d,k} + V_i^{d,k+1} \tag{2b}$$

Here, $V_i^{d,k}$ and $X_i^{d,k}$ are respectively dth element of velocity and position vectors of ith particle at kth iteration, and *Rand1* and *Rand2* 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. (2a), 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 [14].

Kennedy [15] conducted a test to update a particle using Gaussian normal distribution of mean and standard deviation:

$$\mu_i^{d,k} = \frac{(pbest_i^{d,k} + gbest^d)}{2} \tag{3}$$

$$\sigma_i^{d,k} = |pbest_i^{d,k} - gbest^d| \tag{4}$$

$$X_i^{d,k+1} = N(\mu_i^{d,k}, \sigma_i^{d,k}) \tag{5}$$

Here, $\mu_i^{d,k}$ and $\sigma_i^{d,k}$ are respectively the mean and absolute difference of *pbest* and *gbest* in dth dimension of the ith particle at kth iteration. N in Eq. (5) 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. The version of BBPSO in Eq. (5) is known as BB, and, more specifically, BB-*gbest* as *gbest* topology is implemented. Topology is the network of particles in the whole population. *gbest* topology is a fully connected network where the particles are informed of the best position found by the entire population. While *lbest* topology means the population is divided into smaller groups. Best position found in the small group is shared among the members of the smaller group only. Thus, information is passed at slower pace in *lbest* topology than that in *gbest* topology. This has been supported by separate research conducted by van den Bergh and Engelbrecht [16].

In BBPSO, the exploration search is facilitated by $\sigma_i^{d,k}$ term, which tends to zero as the search progresses, and more emphasis will be put on the exploitation of the search. Kennedy [15] proposed an alternative version of BBPSO also, referred to as BBExp in the present paper, where the dth dimension of the ith particle is updated as follows:

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

This means there is a 50% chance that the dth dimension of the ith particle changes to the corresponding *pbest* of the particle. Therefore, particles updated by BBExp algorithm are allowed to take some of the variables in their *pbest*s. Thus, BBExp is said to be biased towards exploiting the *pbest* positions.

3. Development of the proposed BBPSO algorithm

Despite the simplicity and efficiency of BBPSO algorithm, it still suffers from premature convergence [15]. 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. (5) is the value of the particle's *pbest* position itself and the standard deviation is zero. As standard deviation of Gaussian normal distribution becomes zero, *nbest*-index particle (i.e., the particle whose *pbest* is the same as *nbest*) will not be updated. A possible approach to get around this is to assign a small fixed number (such as 0.001) for the standard deviation in updating *nbest*-index particle. However, this implies that the extent of exploration and exploitation for *nbest*-index particle will be constant in the remaining iterations.

In order to have an adaptive balance between exploration and exploitation search nature, the updating approach for *nbest*-index particle has to be modified. One of the strengths of DE is that it is not biased towards any prior defined distribution for sampling mutational step size [12]. Hence, in the proposed BBPSO, mutation and crossover operators of DE algorithm are employed in updating the *nbest*-index particle (and not other particles, which are updated by BBExp algorithm, Eq. (6)), as illustrated by the following equation:

$$X_i^{d,k+1} = \begin{cases} pbest_{i_1}^{d,k} + 0.5 * (pbest_{i_2}^{d,k} - pbest_{i_3}^{d,k}) & \text{if}\, U[0,1] < 0.5 \\ nbest_i^d & \text{otherwise} \end{cases} \quad \text{for}\, i = 1, 2, \ldots, D \tag{7}$$

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 algorithm, 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 of DE. This approach enables the *nbest*-index particle to retain the good variables and has a possibility of changing the non-performing variables.

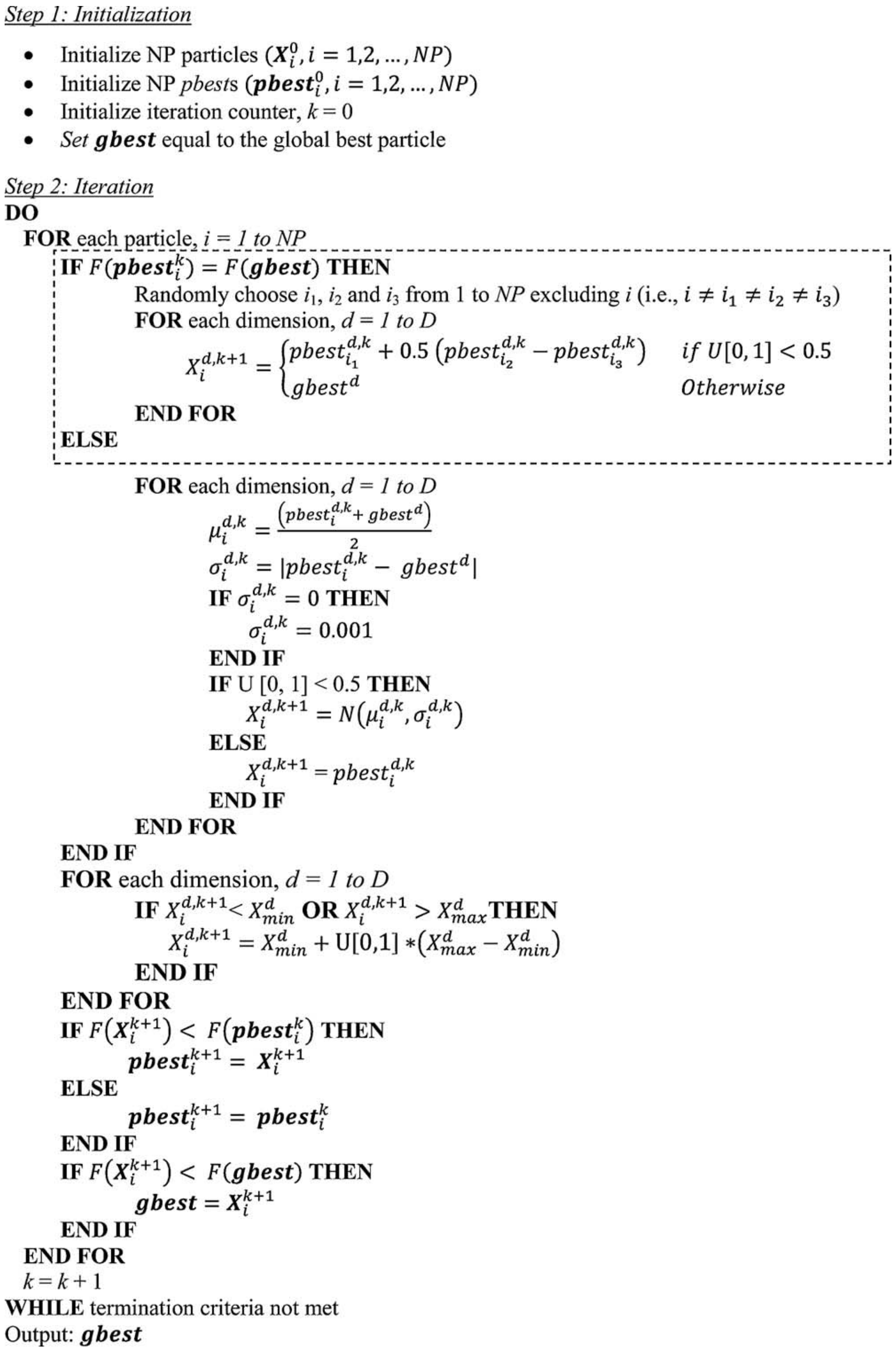
The proposed BBPSO algorithm is hereafter referred to as BBPSO-MC since it employs mutation and crossover operators in some iterations. Its pseudo-code is shown in Fig. 1, with the proposed approach identified in the dashed box. The algorithm checks the objective value of *pbest*; if it is the same as that of *gbest*, the *gbest*-index particle is updated using Eq. (7). Other particles are updated using BBExp algorithm as shown in Eq. (6). If the value of one dimension of the particle happens to be the same as that of *gbest*, then σ is assigned a value of 0.001. This rarely occurs as *gbest*-index particles have been singled out. The pseudo-code in Fig. 1 is based on *gbest* topology; if *lbest* topology is used, *gbest* is simply replaced by *lbest*. Note that boundary violation of a decision variable is handled by:

$$X_i^{d,k+1} = X^d_{\min} + \left(X^d_{\max} - X^d_{\min}\right) * U[0,1] \tag{8}$$

All the algorithms used in this paper are coded in Microsoft Excel Visual Basic Application (VBA). Interested readers can contact the corresponding author for the program.

4. Evaluation of BBPSO-MC on benchmark problems

To illustrate the effectiveness of the proposed method, the performance of the BBPSO-MC algorithm is tested on 10 common benchmark functions (Table 1). There are 9 different 30-D benchmark functions that include five unimodal and four multimodal functions, and one 2-D multimodal function. These were chosen for performance comparison of the BBPSO-MC with the original BBPSO algorithms. In Table 1, *S* and *I*, respectively, stand for search range



```
Step 1: Initialization
  • Initialize NP particles ($\boldsymbol{X}_i^0, i = 1,2,\ldots,NP$)
  • Initialize NP pbests ($\boldsymbol{pbest}_i^0, i = 1,2,\ldots,NP$)
  • Initialize iteration counter, $k = 0$
  • Set $\boldsymbol{gbest}$ equal to the global best particle

Step 2: Iteration
DO
  FOR each particle, $i = 1$ to $NP$
    IF $F(\boldsymbol{pbest}_i^k) = F(\boldsymbol{gbest})$ THEN
          Randomly choose $i_1$, $i_2$ and $i_3$ from 1 to NP excluding $i$ (i.e., $i \neq i_1 \neq i_2 \neq i_3$)
          FOR each dimension, $d = 1$ to $D$
              $X_i^{d,k+1} = \begin{cases} pbest_{i_1}^{d,k} + 0.5\left(pbest_{i_2}^{d,k} - pbest_{i_3}^{d,k}\right) & if\ U[0,1] < 0.5 \\ gbest^d & Otherwise \end{cases}$
          END FOR
    ELSE
          FOR each dimension, $d = 1$ to $D$
              $\mu_i^{d,k} = \frac{\left(pbest_i^{d,k} + gbest^d\right)}{2}$
              $\sigma_i^{d,k} = |pbest_i^{d,k} - gbest^d|$
              IF $\sigma_i^{d,k} = 0$ THEN
                  $\sigma_i^{d,k} = 0.001$
              END IF
              IF $U[0, 1] < 0.5$ THEN
                  $X_i^{d,k+1} = N\left(\mu_i^{d,k}, \sigma_i^{d,k}\right)$
              ELSE
                  $X_i^{d,k+1} = pbest_i^{d,k}$
              END IF
          END FOR
    END IF
    FOR each dimension, $d = 1$ to $D$
          IF $X_i^{d,k+1} < X_{min}^d$ OR $X_i^{d,k+1} > X_{max}^d$ THEN
              $X_i^{d,k+1} = X_{min}^d + U[0,1] * \left(X_{max}^d - X_{min}^d\right)$
          END IF
    END FOR
    IF $F\left(\boldsymbol{X}_i^{k+1}\right) < F\left(\boldsymbol{pbest}_i^k\right)$ THEN
          $\boldsymbol{pbest}_i^{k+1} = \boldsymbol{X}_i^{k+1}$
    ELSE
          $\boldsymbol{pbest}_i^{k+1} = \boldsymbol{pbest}_i^k$
    END IF
    IF $F\left(\boldsymbol{X}_i^{k+1}\right) < F(\boldsymbol{gbest})$ THEN
          $\boldsymbol{gbest} = \boldsymbol{X}_i^{k+1}$
    END IF
  END FOR
  $k = k + 1$
WHILE termination criteria not met
Output: $\boldsymbol{gbest}$
```

Fig. 1. Pseudo-code of BBPSO-MC-*gbest* algorithm.

and initialization range, and I_KM is the different initialization range used in [13]. Initialization range is used only for randomly initializing the particles, and search range is used throughout the optimization run.

The stopping criterion used in testing the 10 benchmark functions is the maximum number of function evaluations, NFE. Each function was solved 30 times, each time using different random number seed, for robust performance analysis. The performance of the proposed algorithm is compared with that of other algorithms based on the mean and standard deviation (Std) of the best objective function values obtained in 30 runs for each benchmark function. The mean value indicates how close the average *gbest* to the global optimum point is and the standard deviation indicates how robust the algorithm is. Robustness of the BBPSO-MC algorithm is also given in terms of success rate, SR (expressed as percentage). Here, a run is considered successful if the *gbest*'s objective function value in a run is within 1.0E−6 of the known global optimum.

Table 1
Details of the benchmark problems used for performance comparison of stochastic optimization methods.

Benchmark function	D	S, I	Global Optimum	f_{min}
*f*1 – *Sphere* $f_1(x)=\sum_{i=1}^{D}x_i^2$	30	$S=I=(-100,100)^D$	[0,. . .,0]	0
*f*2 – *Schwefel's Problem 2.22* $f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30	$S=I=(-10,10)^D$	[0,. . .,0]	0
*f*3 – *Rosenbrock* $f_3(x)=\sum_{j=1}^{D-1}[100(x_i^2-x_{i-1})^2+(x_i-1)^2]$	30	$S=I=(-30,30)^D$	[0,. . .,0]	0
*f*4 – *Step* $f_4(x)=\sum_{i=1}^{D}(\lvert x_i+0.5\rvert)^{2}$	30	$S=I=(-100,100)^D$	[0,. . .,0]	0
*f*5 – *Quadric* $f_5(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_i^2\right)^2$	30	$S=I=(-100,100)^D$	[0,. . .,0]	0
*f*6 – *Schwefel's Problem 2.26* $f_6(x)=-\sum_{i=1}^{D}x_i\sin(\sqrt{x_i})$	30	$S=I=(-500,500)^D$ $I\_KM=(-500,250)^D$	$x_i=20.9687$	−12569.5
*f*7 – *Rastrigin* $f_7(x)=\sum_{i=1}^{D}x_i^2-10\cos(2\pi x_i)+10$	30	$S=I=(-5.12,5.12)^D$ $I\_KM=(2.56,5.12)^D$	[0,. . .,0]	0
*f*8 – *Ackley* $f_8(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	30	$S=I=(-32,32,)^D$ $I\_KM=(16,32)^D$	[0,. . .,0]	0
*f*9 – *Griewank* $f_9(x)=\frac{\sum_{j=1}^{D}x_j^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	$S=I=(-600,600)^D$ $I\_KM=(300,600)^D$	[0,. . .,0]	0
*f*10 – *Camel-back* $f_{10}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$S=I=(-5,5)^D$	0.7126	−1.031 6285

4.1. Results and discussion

In this section, performance results of eight algorithms of BBPSO are presented and discussed. The first four algorithms (BB-*gbest*, BB-*lbest*-2N, BBExp-*gbest* and BBExp-*lbest*-2N) are based on the pseudo-code in [15], while the last four are the BBPSO-MC algorithms using different neighborhood sizes (BBPSO-MC-*gbest*, BBPSO-*lbest*-2N, BBPSO-*lbest*-4N and BBPSO-*lbest*-6N). BBPSO-MC-*gbest* is using the entire population as the neighborhood topology; BBPSO-MC-*lbest*-2N, BBPSO-MC-*lbest*-4N and BBPSO-MC-*lbest*-6N are using neighborhood size of 2, 4 and 6 respectively. All these eight algorithms are coded in *Microsoft Excel 2007* platform, and their performance results for the benchmark functions were obtained in this study.

The eight algorithms are tested on benchmark functions in Table 1, using NP = 20 and NFE = 50,000, and each benchmark function is run 30 times. The results on unimodal functions and multimodal functions are presented in Tables 2 and 3 respectively. The values in bold font in this and other tables are the best values obtained among the different algorithms tried. It can be observed from Table 2 that BB-*gbest* and BBPSO-MC-*lbest*-2N perform relatively better for unimodal functions. Further, *gbest* topology performs better than *lbest* topology for unimodal function. This is due to the nature of *gbest* topology which has higher extent of exploitation than exploration; this is good for unimodal functions in order to have faster convergence. For the five multimodal benchmark functions, BBPSO-MC-*lbest*-2N outperforms BBPSO-MC-*gbest* in terms of mean, standard deviation and SR values except for *f7-Rastrigin* (Table 3). Another observation from Tables 2 and 3 is the general convergence of results from *lbest* topology to *gbest* topology as the size of neighborhood increases. This is clearly illustrated by *f5-Quadric* results in Table 2: as the size of neighborhood increases from 2 to 6 for BBPSO-MC-*lbest*, the mean value moves closer to that obtained by BBPSO-MC-*gbest*. This implies that the size of neighborhood indicates the extent of exploration and exploitation nature of the search. As the neighborhood size increases, the flow of information among the particles in the population is faster and thus results in higher extent of exploitation than exploration.

From all the results in Tables 2 and 3, it can be concluded that BBExp-*gbest* is more robust in terms of SR among the original variants of BBPSO algorithm, for example, 80% SR for *f8-Ackley*. BBExp-*gbest* also shows the lowest mean and standard deviation among the original BBPSO for *f6-Schwefel's Problem 2.26*, *f7-Rastrigin* and *f8-Ackley*. The most efficient and robust topology among the proposed BBPSO-MC algorithms is *lbest*-2N, which

Table 2
Performance of original BBPSO and BBPSO-MC algorithms for unimodal benchmark functions.

Algorithm	*f*1 – Sphere		*f*2 – Schwefel Problem 2.22		*f*3 – Rosenbrock		*f*4 – Step		*f*5 – Quadric	
	Mean (Std)	SR (%)	Mean (Std)	SR (%)	Mean (Std)	SR (%)	Mean (Std)	SR (%)	Mean (Std)	SR (%)
BB-*gbest*	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	44.743427 (38.811398)	0	**0.000000** **(0.000000)**	**100**	**125.593698** **(361.620953)**	**0**
BB-*lbest*-2N	0.000003 (0.000003)	37	0.000110 (0.000444)	0	698.186720 (983.883836)	0	0.000003 (0.000003)	30	34695.473884 (5662.539015)	0
BBExp-*gbest*	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	48.938073 (38.532209)	0	**0.000000** **(0.000000)**	**100**	2377.919098 (2726.607504)	0
BBExp-*lbest*-2N	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	143.077338 (63.077559)	0	**0.000000** **(0.000000)**	**100**	27863.150703 (5236.482128)	0
BBPSO-MC-*gbest*	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	61.482534 (38.404750)	0	**0.000000** **(0.000000)**	**100**	1464.416914 (1530.465404)	0
BBPSO-MC-*lbest*-2N	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	**44.090137** **(28.984722)**	**0**	**0.000000** **(0.000000)**	**100**	18260.042225 (4234.691325)	0
BBPSO-MC-*lbest*-4N	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	51.468557 (34.242224)	0	**0.000000** **(0.000000)**	**100**	17811.661684 (6114.882695)	0
BBPSO-MC-*lbest*-6N	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	47.904903 (38.532255)	0	**0.000000** **(0.000000)**	**100**	15131.511286 (4461.860694)	0

has the highest SR across all the benchmark functions tested and gives the lowest mean value for *f6-Schwefel's Problem 2.26* and *f3-Rosenbrock*. Comparing BBExp-*gbest* and BBPSO-MC-*lbest*-2N, it can be concluded that the latter performs better for majority of the benchmark functions, especially for the multimodal functions.

5. Parameter estimation problems in VLE data modeling

To show the effectiveness of the proposed method, the proposed BBPSO-MC algorithm is applied to estimate parameters in the 16 vapor–liquid equilibrium (VLE) models to fit the experimental data. Parameter estimation based on both least squares (LS) and error-in-variable (EIV) approaches are considered. Of these, 10 problems are based on LS approach and another 6 problems are based on VLE-EIV approach; both these approaches minimize an objective function subject to constraints arising from model equations [2–4,17,18]. Recently, our group solved these parameter estimation problems by PSO and other stochastic global optimization algorithms [6,19]. All these are employed in the present study too. The performance results of BBPSO-MC algorithms are compared with those for simulated annealing (SA), genetic algorithm (GA), differential evolution (DE) and DE with taboo list (DETL) techniques.

5.1. VLE models and parameter estimation

5.1.1. Least squares approach

Consider a set of observations q_{ij} of $i = 1, 2, \ldots, m$ dependent/response variables from $j = 1, 2, \ldots, ndat$ experiments, where the responses can be expressed by an explicit model $f_i(r_j, \theta)$ with nl independent variables $r_j = (r_{1,j}, \ldots, r_{nl,j})$ and $npar$ parameters $\theta = (\theta_1, \ldots, \theta_{npar})$. Measurement errors in r_j are neglected in LS approach. The details of VLE-LS data are showing in Table 7. There are many different objective functions that can be used to estimate the parameter values that provide the best fit for a model [20]. In this study, the objective function involving fractional errors shown in Eq. (9) is used since it tends to weigh errors in small and large quantities equally [6].

$$F_{obj} = \sum_{j=1}^{ndat} \sum_{i=1}^{m} \left(\frac{q_{ij} - f_i(r_j, \theta)}{q_{ij}} \right)^2 \qquad (9)$$

The decision variables in the LS approach are $npar$ parameters θ.

In the case of VLE data, excess Gibbs energy equation is normally used for phase equilibrium modeling. Hence, the objective function used in Eq. (9) can be modified as follows to fit the activity

Table 3
Performance of original BBPSO and BBPSO-MC algorithms for multimodal benchmark functions.

Algorithm	f6-Schwefel problem 2.26		f7-Rastrigin		f8-Ackley		f9-Griewank		f10-Camel-back	
	Mean (Std)	SR (%)	Mean (Std)	SR (%)	Mean (Std)	SR (%)	Mean (Std)	SR (%)	Mean (Std)	SR (%)
BB-*gbest*	−8859.773429 (422.404766)	0	116.807496 (30.088388)	0	9.061415 (9.397339)	30	0.017614 (0.025752)	40	**−1.031628** **(0.000000)**	**100**
BB-*lbest*-2N	−7101.224389 (660.220007)	0	104.199487 (23.237132)	0	10.415084 (9.185170)	0	**0.000000** **(0.000000)**	**93**	**−1.031628** **(0.000000)**	**100**
BBExp-*gbest*	−9956.105226 (541.976212)	0	17.378619 (7.077632)	0	3.770410 (7.735386)	80	0.002378 (0.006562)	83	**−1.031628** **(0.000000)**	**100**
BBExp-*lbest*-2N	−8477.968148 (357.647651)	0	38.897715 (11.380080)	0	3.888472 (7.110153)	0	0.006020 (0.008814)	20	**−1.031628** **(0.000000)**	**100**
BBPSO-MC-*gbest*	−9802.390994 (447.325859)	0	**16.350482** **(5.124742)**	0	1.365896 (5.045482)	90	0.001479 (0.003579)	83	**−1.031628** **(0.000000)**	**100**
BBPSO-MC-*lbest*-2N	**−10668.638607** **(757.750508)**	0	73.876299 (11.187710)	0	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	**−1.031628** **(0.000000)**	**100**
BBPSO-MC-*lbest*-4N	−10493.043474 (559.226575)	0	67.982139 (12.81535)	0	**0.000000** **(0.000000)**	**100**	**0.000000** **(0.000000)**	**100**	**−1.031628** **(0.000000)**	**100**
BBPSO-MC-*lbest*-6N	−10627.655795 (360.021301)	0	46.460206 (20.996651)	0	1.306393 (4.971652)	93	0.000000 (0.000001)	97	**−1.031628** **(0.000000)**	**100**

Table 4
Details of VLE data modeling problems used for performance comparison of stochastic optimization methods, in the classical least squares (LS) formulation.

LS no.	System	Data	Models	Decision variables	Global optimum	Remarks
LS-1		$P = 100$ mm Hg $ndat = 9$			$F_{obj} = 0.01026$ $\theta_1 = -567.96$ $\theta_2 = 745.33$	
LS-2	Tert butanol + 1 butanol	$P = 700$ mm Hg $ndat = 9$	Wilson and ideal gas model	$D = 2$ $\theta_1, \theta_2 \in (-8500, 320{,}000)$	$F_{obj} = 0.013690$ $\theta_1 = -733.95$ $\theta_2 = 1318.23$	
LS-3		$P = 500$ mm Hg $ndat = 9$			$F_{obj} = 0.006852$ $\theta_1 = -718.01$ $\theta_2 = 1264.74$	2 minima [15–17]
LS-4			Wilson and ideal gas model	$D = 2$ $\theta_1, \theta_2 \in (-8500, 320{,}000)$	$F_{obj} = 1.039134$ $\theta_1 = 5072.361$ $\theta_2 = -1921.62$	
LS-5	Water + 1,2 ethanediol	$P = 430$ mm Hg $ndat = 18$	UNIQUAC and ideal gas model	$D = 2$ $\theta_1, \theta_2 \in (-5000, 20{,}000)$	$F_{obj} = 1.408547$ $\theta_1 = -1131.84$ $\theta_2 = 3617.65$	
LS-6			NRTL and ideal gas model	$D = 3$ $\theta_1, \theta_2 \in (-2000, 5000)$ $\alpha_{12} \in (0.01, 10.0)$	$F_{obj} = 1.253531$ $\theta_1 = -678.99$ $\theta_2 = 3046.13$ $\alpha_{12} = 0.621375$	
LS-7		$T = 50\,^{\circ}$C $ndat = 11$			$F_{obj} = 0.008935$ $\theta_1 = -424.08$ $\theta_2 = 983.06$	
LS-8	Benzene + hexafluorobenzene	$P = 300$ mm Hg $ndat = 17$	Wilson and ideal gas model	$D = 2$ $\theta_1, \theta_2 \in (-8500, 320{,}000)$	$F_{obj} = 0.014860$ $\theta_1 = -432.49$ $\theta_2 = 992.85$	2 minima [15,17,18]
LS-9		$P = 760$ mm Hg $ndat = 29$			$F_{obj} = 0.014616$ $\theta_1 = -334.70$ $\theta_2 = 704.74$	
LS-10		$T = 30\,^{\circ}$C $ndat = 10$			$F_{obj} = 0.011783$ $\theta_1 = -467.76$ $\theta_2 = 1313.94$	

Table 5
Details of VLE data modeling problems used for performance comparison of stochastic optimization methods, in the error-in-variable (EIV) formulation.

EIV no.	System	Data	Models	Decision variables	Global optimum	Remarks
EIV-1		$P = 500$ mmHg $ndat = 16$ σ (0.003, 0.0029,1.7, 0.083)		$D = 34$ θ_1, $\theta_2 \in (-10{,}000, 200{,}000)$ $x^t_{ij} \in (x_{ij} - 3\sigma, x_{ij} + 3\sigma)$ $T^t \in (T - 3\sigma, T + 3\sigma)$	$F_{obj} = 19.998720$ $\theta_1 = -429.85$ $\theta_2 = 1029.32$	
EIV-2		$P = 300$ mmHg $ndat = 17$ σ (0.003, 0.0029,1.7, 0.083)		$D = 36$	$F_{obj} = 42.343724$ $\theta_1 = -437.72$ $\theta_2 = 1003.12$	
EIV-3	Benzene + hexafluorobenzene	$T = 30\,^{\circ}Cndat = 10$ σ (0.001, 0.01,0.75, 0.1)	Wilson and ideal gas model	$D = 22$	$F_{obj} = 11.898795$ $\theta_1 = -472.00$ $\theta_2 = 1274.32$	2 minima [17,19,20]
EIV-4		$T = 40\,^{\circ}Cndat = 10$ σ (0.001, 0.01,0.75, 0.1)		$D = 22$	$F_{obj} = 11.170496$ $\theta_1 = -462.51$ $\theta_2 = 1197.31$	
EIV-5		$T = 50\,^{\circ}Cndat = 11$ σ (0.003, 0.0029,1.7, 0.083)		$D = 24$	$F_{obj} = 25.671042$ $\theta_1 = -415.76$ $\theta_2 = 944.57$	
EIV-6		$T = 60\,^{\circ}Cndat = 10$ σ (0.003, 0.0029,1.7, 0.083)		$D = 22$	$F_{obj} = 19.401593$ $\theta_1 = -437.05$ $\theta_2 = 1065.04$	

Table 6
Success rate (SR) of PS, BBPSO-MC-lbest and BBPSO-MC-*gbest* for VLE-LS problems using SC-1 with NP of 10D and 50D at 5000 and 10,000 NFEs.

LS	PS				BBPSO-MC-*lbest*				BBPSO-MC-*gbest*			
no.	NFE = 5000		NFE = 10,000		NFE = 5000		NFE = 10,000		NFE = 5000		NFE = 10,000	
	10D	50D	10D	50D	10D	50D	10D	50D	10D	50D	10D	50D
LS-1	20	3	23	2	64	34	73	75	59	65	65	74
LS-2	1	0	4	2	80	21	87	85	76	74	78	91
LS-3	81	0	86	0	81	38	86	90	52	73	64	84
LS-4	27	0	30	0	83	56	87	97	56	72	62	80
LS-5	34	43	42	57	68	94	71	98	59	60	63	70
LS-6[1]	93	99	94	100	98	100	99	100	81	91	84	97
LS-7	0	1	3	1	83	41	91	91	76	73	86	90
LS-8	1	1	5	2	89	50	99	81	83	76	91	92
LS-9	4	2	4	3	92	56	100	99	96	90	100	98
LS-10	4	0	6	0	91	32	99	96	83	76	86	99
GSR	27	15	30	17	83	52	89	91	72	75	80	88

[1] NFE = 7500 at NP = 10D and NFE = 15,000 at NP = 50D as LS-6 problem has 3 decision variables. All other LS problems have 2 decision variables only.

coefficient data:

$$F_{obj} = \sum_{j=1}^{ndat}\sum_{i=1}^{c}\left(\frac{\gamma_{i,j}^{\exp}-\gamma_{i,j}^{calc}}{\gamma_{i,j}^{\exp}}\right)^{2} \qquad (10)$$

where $\gamma_{i,j}^{\exp}$ is the experimental value for the activity coefficient of component i in jth experiment, $\gamma_{i,j}^{calc}$ is the calculated value for the activity coefficient of component i in jth experiment, and c is the number of components in the mixture. The assumption behind Eq. (10) is that the standard error in the measurement of γ_{ij} is proportional to its value [1]. $\gamma_i^{\exp}$ can be calculated from VLE data as follows:

$$\gamma_i^{\exp} = \frac{y_i^{\exp}P}{x_i^{\exp}P_i^{o}} \quad i = 1, \ldots, c \qquad (11)$$

where $x_i^{\exp}$ is the experimental mole fraction of component i in liquid at equilibrium, $\boldsymbol{y}_i^{\exp}$ the experimental mole fraction of component i in vapor phase at equilibrium, $\boldsymbol{P}_i^{0}$ is the vapor pressure of pure component i at the system temperature **T**, and **P** is the pressure of the system. For Eq. (11), it is assumed that, at low pressure, the fugacity coefficients of pure components cancel each other and the values of Poynting corrections are very close to one [6].

Non-Random-Two-Liquid (NRTL), Wilson and UNIversal QUAsi-Chemical (UNIQUAC) models have been used to evaluate γ_i^{calc} and Eq. (10) is optimized by changing the energy parameters (θ) of these models. Owing to the explicit nature of the equations for γ_i^{calc}, global minimization of LS objective function, Eq. (9) can be done as an unconstrained optimization problem.

5.1.2. Error-in-variable approach

Unlike LS approach, EIV approach considers the error in the state variables $Z_{i,j}$ for the experiments of the system to be modeled (namely, **x**, **y**, **T** and **P** for VLE data modeling). The details of VLE-EIV data are showing in Table 8. This results in the following objective function:

$$F_{obj} = \sum\nolimits_{j=1}^{ndat}\sum\nolimits_{i=1}^{nest}\frac{z_{i,j}^{t}-z_{i,j}}{\sigma_i^2} \qquad (12)$$

$$\text{subject to}\, \boldsymbol{g}(Z_{i,j}^{t}, \boldsymbol{\theta}) = 0 \quad i = 1, \ldots, nest \quad j = 1, \ldots, ndat \qquad (13)$$

Here, $\boldsymbol{g}$ is a vector of model functions, *nest* is the number of state variables, $z_{i,j}^{t}$ is the unknown "true" value of ith state variable in jth experiment and σ_i is the standard deviation associated with the measurement of ith state variable. The decision variables of EIV formulation include the set of $z_{i,j}^{t}$ values in addition to *npar* parameters $\boldsymbol{\theta}$ in the model. This leads to significant increase in the dimensionality of the optimization problem depending on the number of experimental data. The bounds for the state variables are usually three standard deviations away from the measured values.

Assuming the experiment is conducted at low pressure, VLE problems can be defined by the following equations:

$$\boldsymbol{P} = \sum\nolimits_{i=1}^{c}(\boldsymbol{\gamma}_i * \boldsymbol{x}_i * \boldsymbol{P}_i^{0}) \qquad (14)$$

$$\boldsymbol{y}_i = \frac{\gamma_i * x_i * P_i^{0}}{\sum_{j=1}^{c}(\gamma_j * x_j * P_j^{0})} \quad i = 1, \ldots, c \qquad (15)$$

The above equations allow the VLE problems to be solved as an unconstrained optimization problem through their substitution for $\boldsymbol{P}^{t}$ and $\boldsymbol{y}_i^{t}$ in the EIV objective function equation (12) to give the following objective function for EIV problem formulation.

$$F_{obj} = \sum\nolimits_{j=1}^{ndat}\sum\nolimits_{i=1}^{c}\left[\frac{(x_{i,j}^{t}-x_{i,j}^{\exp})^2}{\sigma_{x_i}^2}+\frac{(y_{i,j}^{t}-y_{i,j}^{\exp})^2}{\sigma_{y_i}^2}+\frac{(T_j^{t}-T_j^{\exp})^2}{\sigma_T^2}+\frac{(P_j^{t}-P_j^{\exp})^2}{\sigma_P^2}\right] \qquad (16)$$

The independent variables are the set of $\boldsymbol{z} = (\boldsymbol{x}_i^{\exp}, \boldsymbol{T}^{\exp})$ for all the measurements. The decision variables for optimization are the set of $\boldsymbol{z}^{t} = (\boldsymbol{x}_i^{t}, \boldsymbol{T}^{t})$ and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{npar})$, and the total number of decision variables is $npar + c \times ndat$. Both LS and EIV formulations involve complex, non-linear thermodynamic models, which results in highly non-linear and potentially non-convex objective function with local optima.

5.2. Test examples and evaluation procedure

Altogether there are 16 VLE problems, consisting of ten and six problems based on LS and EIV approaches respectively. The details of these problems can be found in Tables 4 and 5. All these problems are multimodal with number of decision variables ranging from 2 to 36. Each VLE problem is solved 100 times independently except for VLE-EIV-1 and VLE-EIV-2 problems, which were solved only 50 times due to the significantly large computational time for each run. The performances of stochastic algorithms are compared based on SR and NFE for two stopping criteria: SC-1 based on the maximum number of iterations and SC-2 based on the maximum number of iterations (SC_{max}) without improvement in the *gbest* objective function value. SC-2 is also known as *ImpBest*, an improvement-based stopping criterion. A trial/run is considered successful if the *gbest* objective function value obtained is within 1.0E−5 from the

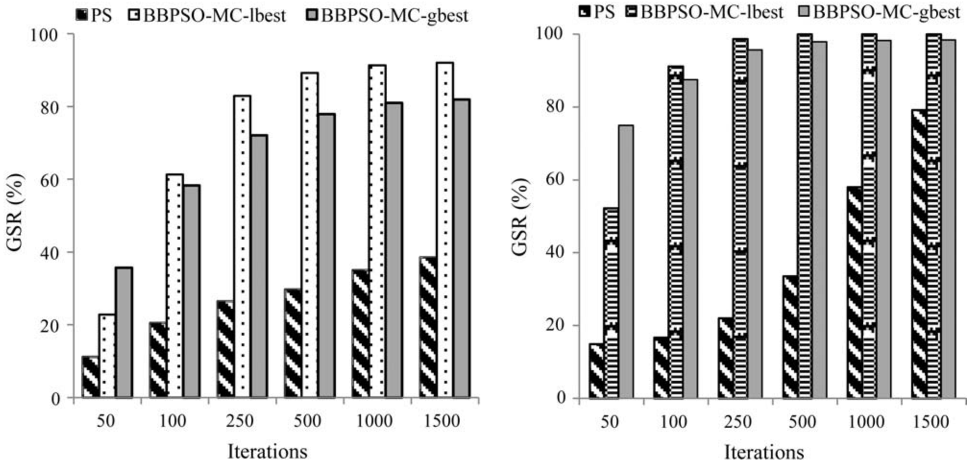


Fig. 2. Global success rate (GSR) versus Iteration of VLE-LS problems using SC-1for PS, BBPSO-MC-*lbest* and BBPSO-MC-*gbest*: (a) NP is 10D and (b) NP is 50D.

known global optimum. Also, global success rate (GSR) of different algorithms is reported for the LS and EIV problems. GSR is defined as

$$GSR = \sum_{i=1}^{np} \frac{SR_i}{np} \tag{17}$$

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

In total, three algorithms were tried for solving the VLE problems; they are two proposed BBPSO algorithms (namely, BBPSO-MC-*gbest* and BBPSO-MC-*lbest* with 2 neighbors) and the PSO algorithm used in [6] which is simply referred to as PS algorithm in this paper. Later, results of the best algorithm among these three algorithms are compared with those by the other stochastic algorithms reported in [6].

At the end of the 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. Since the PSO algorithms are implemented in Excel platform, *Solver* tool is chosen as the local optimizer. It uses generalized reduced gradient algorithm which needs good initial guess; here, *gbest* of the population at the end of the stochastic algorithm is used as the initial guess. *Solver* is likely to locate the global optimum if the initial guess, *gbest*, is in the global optimum region. During the course of this study, we found that default settings of *Solver* may lead to premature convergence, particularly for VLE-EIV problems. Use of central difference (instead of forward difference) approximation for derivatives overcomes this difficulty thus improving convergence to the global minimum. The parameter estimation results reported in this paper were obtained using the following settings for *Solver* – Precision: 0.0000001, Tolerance: 5%, Convergence: 0.0000001, Estimates: Tangent, Derivatives: Central and Search: Newton.

5.3. Results and discussion

5.3.1. Using least squares approach

For the 10 VLE-LS problems, GSR values of all VLE-LS problems by different algorithms using SC-1 with NP of 10D and 50D respectively are illustrated in Fig. 2. The results are collected at different iteration levels starting from 50- to 1500-iteration level. As expected, Fig. 2 indicates clearly that GSR improves with increasing number of iterations. For example, for LS problem, BBPSO-MC-*gbest* with 10D gives GSR of 23% at 50-iteration level but this value increases to 92% at 1500-iteration level. Furthermore, the effect of increasing the number of iterations on GSR is more significant for BBPSO algorithms than for PS algorithm. Also, as NP increases from 10D in Fig. 2(a) to 50D in Fig. 2(b), GSR increases for all the algorithms tested across different iteration levels. However, this is at the expense of computational resources as higher NP requires higher NFE.

As can be seen from Figs. 2(a) and (b), BBPSO algorithms achieve GSR close to 100% for most of the VLE-LS problems at high iteration levels. Among the three algorithms with NP of 10D, BBPSO-MC-*lbest* gives better results across different VLE-LS problems and different iteration levels. BBPSO-MC-*gbest* algorithm is also very competitive as it manages to achieve high GSR at a very low iteration level of 50. As the NP is increased to 50D, BBPSO-MC-*lbest* is still better than the remaining two BBPSO algorithms. Also, at NP of 50D, GSR close to 100% is observed at iteration level as low as 500 for the proposed BBPSO algorithms (Fig. 2(b)). Thus, it is not necessary to continue running the algorithms at higher iteration levels which can be computationally expensive. Comparing PS and proposed BBPSO algorithms, it can be concluded that the latter are more robust and efficient. In general, proposed BBPSO algorithms give better results than PS algorithm for all VLE-LS problems tested using different NP and iteration levels.

The effect of population size on VLE-LS problems is studied at NFE of 5000 and 10,000 using NP of 10D and 50D using three methods. The results are compiled in Table 6. It can be concluded that NP has different effect on different algorithms. PS and BBPSO-MC-*lbest* algorithms seem to work marginally better at low NP. On the other hand, BBPSO-MC-*gbest* works marginally better at high NP. Overall, the effect of NP is not so significant for VLE-LS problems.

The previous discussion on VLE-LS problems are based on SC-1. In general, additional iterations with no significant improvement in the results are waste of computational resources. Thus, it is essential for the optimization algorithm to stop at the right time incurring least computational resources without compromising reliability of finding the global optimum. As such, in order to study the effect of stopping criteria on different algorithms, SC-2 is used. Tables 7 and 8 summarize the results obtained by different algorithms using SC-2 with $SC_{max} = 6D$ and $SC_{max} = 12D$ along with maximum allowable iterations of 1500, using NP of 10D and 50D respectively. Tables 7 and 8 suggest that the higher the SC_{max}, the better the results are for VLE-LS problems but require higher NFE values. As the algorithms are allowed to run for a longer duration, the probability of them locating the global optimum region is increased. From the results, it is clear that the performance of BBPSO-MC is significantly better than PS algorithm. BBPSO-MC-*lbest* seems to perform better as it obtains a comparable SR at relatively lower NFE than BBPSO-MC-*gbest* and PS across different

Table 7
Success rate (SR) of PS, BBPSO-MC-*lbest* and BBPSO-MC-*gbest* for VLE-LS problems using SC-2 with NP of 10D and maximum iteration level of 1500.

LS No.	SC_{max}	PS		BBPSO-MC-*lbest*		BBPSO-MC-*gbest*	
		SR	NFE	SR	NFE	SR	NFE
LS-1	6D	4	1280	11	1356	**28**	2040
	12D	15	3267	**46**	3298	**46**	3679
LS-2	6D	1	360	9	1313	**45**	2403
	12D	1	600	56	5659	**59**	4241
LS-3	6D	8	1315	8	1315	**35**	2555
	12D	**50**	5377	**50**	5377	47	4735
LS-4	6D	2	3400	10	1740	**32**	2058
	12D	13	3243	**50**	3569	42	2982
LS-5	6D	11	504	34	668	**38**	1480
	12D	15	1104	**49**	1846	39	2532
LS-6	6D	72	1180	**82**	1796	76	5996
	12D	86	2757	**96**	3728	79	11,012
LS-7	6D	0	NA[2]	9	1384	**36**	2652
	12D	0	NA	49	5434	**55**	5157
LS-8	6D	0	NA	9	1189	33	2839
	12D	0	NA	47	4317	**55**	5304
LS-9	6D	1	720	14	1097	**36**	2066
	12D	2	1270	**59**	2500	51	4803
LS-10	6D	0	NA	6	1143	**38**	2621
	12D	0	NA	59	5233	66	5121

2 – Zero SR has been reported for this algorithm on the problem.

problems. This is best illustrated by LS-1 problem using NP = 10D and SC_{max} = 12D, where BBPSO-MC-*lbest* obtains 46% SR with 3298 NFE while BBPSO-MC-*gbest* obtains 46% SR with NFE of 3679 and PS obtain 15% SR with NFE of 3267 (Table 7). This shows that BBPSO-MC is more efficient than PS.

In order to compare the different algorithms with different stopping criterion, Fig. 3 summarizes GSR of three algorithms with three stopping criteria. It can be observed that the use of SC-2 gives lower GSR value compared to SC-1. This is due to the nature of the stopping criteria where SC-2 will force the algorithm to stop when the best particle in the population fails to improve after a certain number of iterations. Thus algorithms which tend to be trapped in the local optima easily are stopped prematurely. This is clearly illustrated in Fig. 3, where the GSR of proposed BBPSO algorithms is consistently higher at higher SC_{max}. This indicates that once the PS algorithm is trapped, it is hard for the population to escape from the local optimum.

By comparing Fig. 3(a) and (b), it can be concluded that the performance gap in terms of GSR between the use of SC-1 and SC-2 is significantly narrowed when NP increases from 10D to 50D. This implies that the use SC-2 at higher NP is desirable as the bare-bones algorithms can maintain high GSR at significantly lower NFE. Overall, similar to the conclusion drawn previously, BBPSO algorithms outperform PS algorithm for most of the VLE-LS problems using SC-2 at different NPs and different iteration levels.

5.3.2. *Using error-in-variable approach*

Unlike LS approach, EIV approach considers errors in the measured data, and hence the number of decision variables, $npar + c \times ndat$, increases significantly. This implies that VLE-EIV problems are more challenging than VLE-LS problems. Fig. 4(a) presents GSR results using SC-1 as the stopping criterion at different iteration levels; these are better than those in Fig. 2 at the corresponding iteration level, probably due to large D (22–36) and consequently NP (=10D). Similar to the previous discussion on LS approach, the higher the iteration level, the better the results are. Increasing the iteration level enhances performance of BBPSO algorithms to a greater extent than PS algorithm. BBPSO-MC-*lbest*,

Table 8
Success rate (SR) and NFE of PS, BBPSO-MC-*lbest* and BBPSO-MC-*gbest* for VLE-LS problems using SC-2 with NP of 50D and maximum iteration level of 1500.

LS no.	SC_{max}	PS		BBPSO-MC-*lbest*		BBPSO-MC-*gbest*	
		SR	NFE	SR	NFE	SR	NFE
LS-1	6D	0	NA	27	5437	**66**	8283
	12D	3	5067	62	9094	**84**	16,548
LS-2	6D	1	2400	13	5862	**76**	9499
	12D	0	NA	65	10,524	**94**	20,032
LS-3	6D	1	260	20	5535	**75**	10,159
	12D	**0**	NA	82	9462	**91**	19,945
LS-4	6D	2	2100	27	5530	**67**	8094
	12D	1	2700	**84**	11,268	76	10,068
LS-5	6D	31	2632	**75**	6538	45	6538
	12D	49	5355	**97**	7820	52	8998
LS-6	6D	98	6444	**100**	7761	96	23,095
	12D	**100**	13,683	100	17,403	99	29,662
LS-7	6D	1	1800	34	5488	**65**	11,032
	12D	1	3000	79	9975	**91**	20,069
LS-8	6D	0	NA	29	5355	79	10,809
	12D	1	4800	71	8380	**96**	19,994
LS-9	6D	4	1850	39	4662	**70**	9451
	12D	4	3050	86	8117	**88**	21,538
LS-10	6D	0	NA	21	6319	**77**	11,038
	12D	0	NA	78	12,822	**97**	17,192

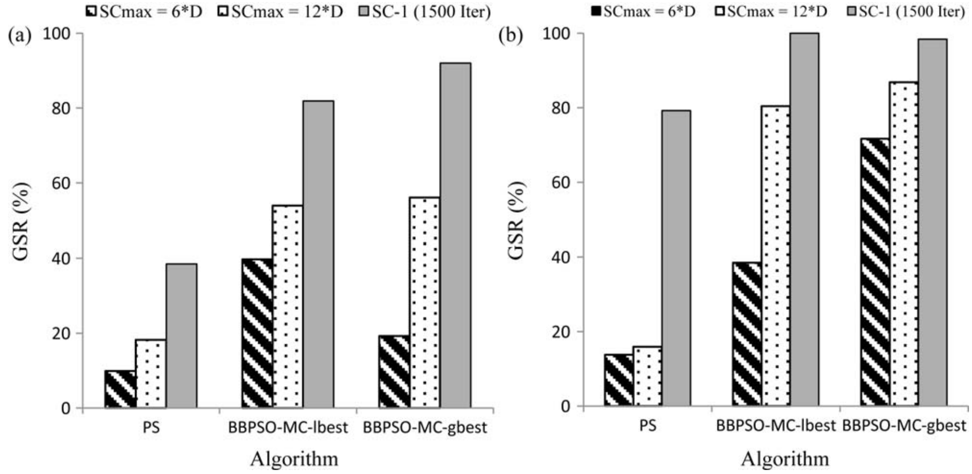


Fig. 3. Global success rate (GSR) of different algorithms for VLE-LS problems using SC-2 (SC_{max} = 6D and SC_{max} = 12D) and SC-1 (1500 iterations): (a) NP is 10D and (b) NP is 50D.

which shows good results for VLE-LS problems, has slightly worse performance in VLE-EIV problems compared to BBPSO-MC-*gbest*. Overall, BBPSO-MC-*gbest* and BBPSO-MC-*lbest* algorithms are more robust and efficient than PS algorithm at all iteration levels, as shown in Fig. 4(a).

Table 9 gives the results of VLE-EIV problems using SC-2 as the stopping criterion. It is obvious that the proposed BBPSO obtained higher SR compared with PS. BBPSO-MC-*gbest* gives SR of 100% for all the VLE-EIV problems. Thus, the proposed BBPSO-MC-*gbest* maintains its superiority over PS algorithm for solving VLE-EIV problems using SC-2 as the stopping criterion. The average NFE of successful runs, reported in Table 9, indicates that BBPSO-MC-*gbest* has slightly better SR and requires fewer NFE compared to BBPSO-MC-*lbest* algorithm. The performance comparison of different algorithms with different stopping criterion is reported in Fig. 4(b), which shows that SC-1 gives only slightly higher GSR across different problems for PS algorithm and the proposed BBPSO algorithms. This indicates that SC-2 is good stopping criterion for VLE-EIV problems. Thus, BBPSO algorithms outperform PS algorithm for most of the VLE-EIV problems.

5.3.3. *Comparison of BBPSO with other stochastic algorithms*

Of the many stochastic algorithms, SA, DE, DETL and GA have been evaluated for parameter estimation involved in modeling VLE data in [6]. 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. In view of the premature convergence observed with the default settings of Solver, we re-examined the results on VLE modeling problems in [6]. However, no option for approximating numerical derivatives by central differences is available in DBCONF. So, DBCONF was replaced by Excel Solver with the settings stated above (namely, Precision: 0.0000001, Tolerance: 5%, Convergence: 0.0000001, Estimates: Tangent, Derivatives: Central and Search: Newton), and SA, DE and DETL along with Solver were tried again for the VLE data modeling problems considered in this work. This was facilitated as we had the best solutions at the end of each stochastic algorithm tried in [6]. These new results for SA, DE and DETL were better, particularly for VLE-EIV problems, than those in [6], which indicates the need for extra care in using a local optimizer. In this study, the new results were used for comparison with BBPSO and PS algorithms.

Fig. 5(a) and (b) shows GSR values for VLE-LS problems using NP of 10D and 50D respectively. It can be seen that BBPSO-MC-*lbest* and BBPSO-MC-*gbest* achieve consistently higher GSR at different iteration levels and for different NP. In fact, BBPSO-MC-*lbest* results at 250-iteration level are already superior to those of other stochastic algorithms at higher iteration levels. This is a clear indication

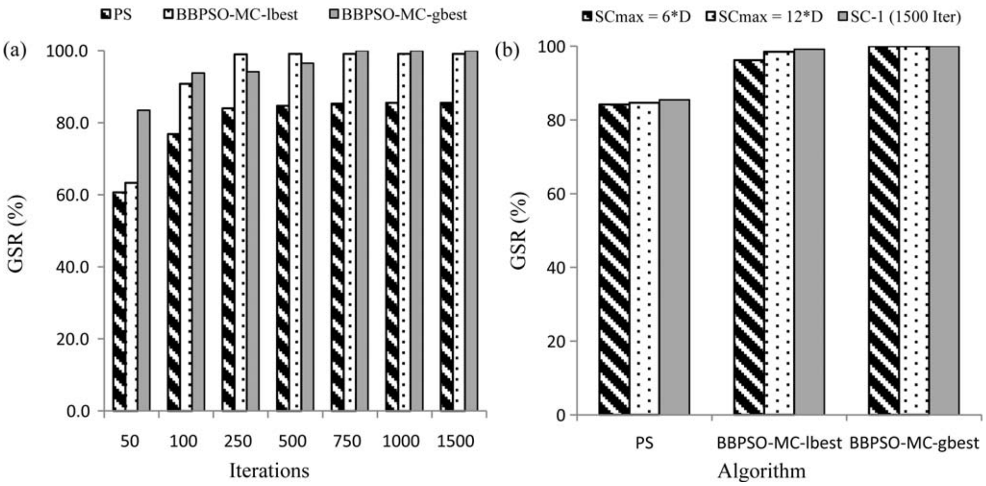


Fig. 4. (a) Global success rate (GSR) versus iteration of PS, BBPSO-MC-*lbest* and BBPSO-MC-*gbest* for VLE-EIV problems. b) Global Success Rate (GSR) versus different algorithms (PS, BBPSO-MC-*lbest* and BBPSO-MC-*gbest*) of VLE-EIV problems using SC-2 (SC_{max} = 6D and SC_{max} = 12D) and SC-1(1500 iterations).

Table 9
Success rate (SR) and NFE of PS, BBPSO-MC-*lbest* and BBPSO-MC-*gbest* for VLE-EIV problems using SC-2 with NP of 10D and maximum iteration level of 1500.

EIV no.	SC_{max}	PS		BBPSO-MC-*lbest*		BBPSO-MC-*gbest*	
		SR	NFE	SR	NFE	SR	NFE
EIV-1	6D	98	185,251	**100**	510,000	**100**	510,000
	12D	98	316,908	**100**	510,000	**100**	510,000
EIV-2	6D	96	199,335	**100**	540,000	**100**	416,974
	12D	96	353,010	**100**	540,000	**100**	491,414
EIV-3	6D	52	88,355	89	308,551	**100**	166,806
	12D	52	159,315	**100**	330,000	**100**	202,655
EIV-4	6D	64	92,850	**100**	229,275	**100**	209,581
	12D	67	156,791	**100**	324,449	**100**	252,657
EIV-5	6D	97	99,523	96	354,993	**100**	198,643
	12D	97	171,845	96	358,583	**100**	238,493
EIV-6	6D	98	89,688	92	316,362	**100**	182,035
	12D	98	151,546	95	328,223	**100**	232,146

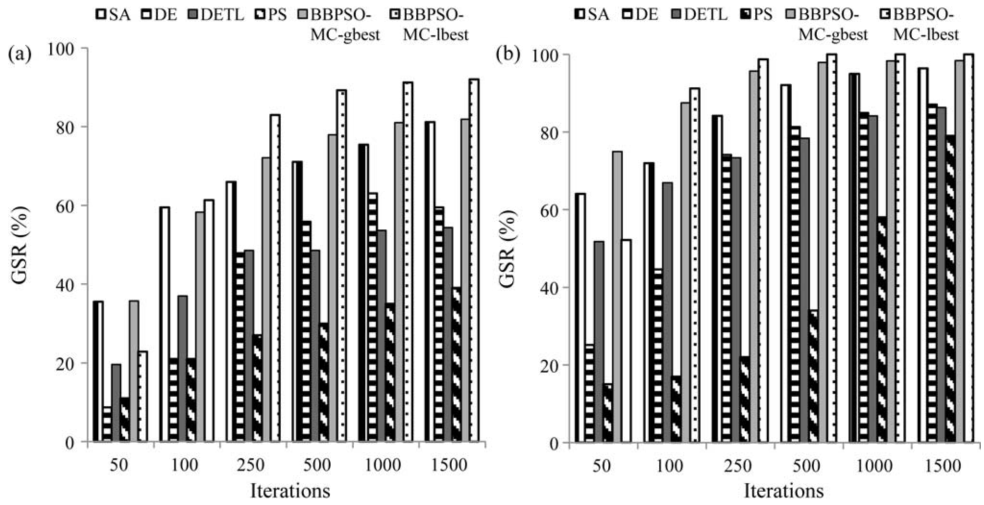


Fig. 5. Global success rate (GSR) versus iteration of SA, DE, DETL, PS BBPSO-MC-*gbest* and BBPSO-MC-*lbest* for VLE-LS problems using SC-1: (a) NP is 10D and (b) NP is 50D.

that BBPSO-MC-*lbest* is very efficient and robust for solving VLE-LS problems. Further, PS results are inferior to those of BBPSO-MC-*lbest*. For VLE-EIV problems, BBPSO-MC-*lbest* and BBPSO-MC-*gbest* outperform all other algorithms tested (Fig. 6); BBPSO-MC-*lbest* performance for VLE-EIV problems is not as good as that of BBPSO-MC-*gbest* at low iteration levels. Both the BBPSO-MC algorithms seem to be more efficient than SA, DE, DETL and PSO algorithms to achieve the same GSR for VLE-EIV problems.

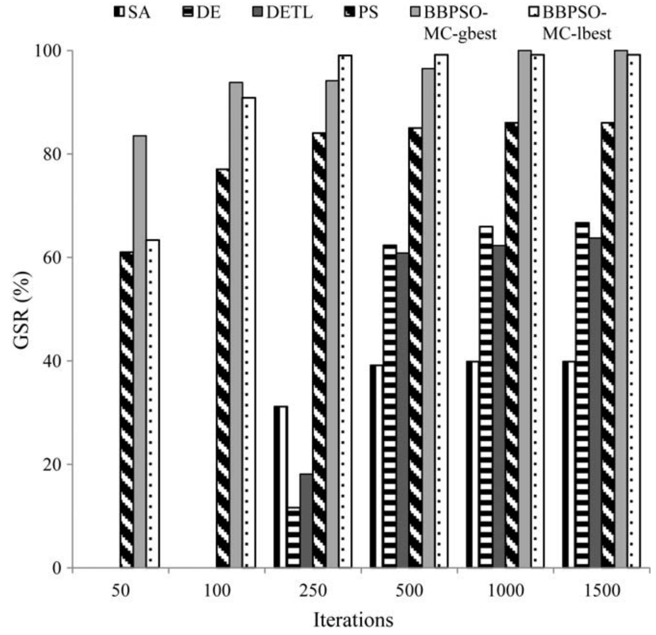


Fig. 6. Global success rate (GSR) versus iteration of SA, DE, DETL, PS BBPSO-MC-*gbest* and BBPSO-MC-*lbest* for VLE-EIV problems using SC-1 with NP at 10D.

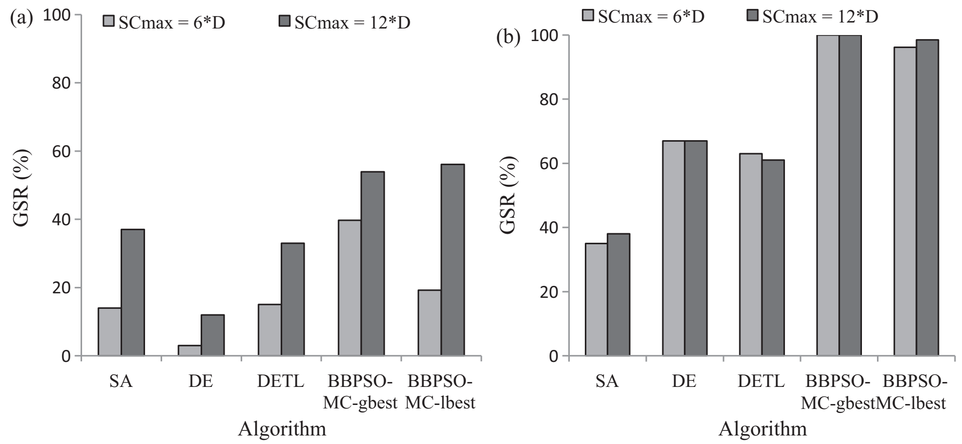


Fig. 7. Global success rate (GSR) versus different algorithm (SA, DE, DETL, BBPSO-MC-*gbest* and BBPSO-MC-*lbest*) using SC-2 (SC_{max} = 6D and SC_{max} = 12D) with NP of 10D: (a) VLE-LS problems and (b) VLE-EIV problems.

As depicted in Fig. 7, performance of the stochastic algorithms tested generally deteriorates when SC-2 is used as the stopping criterion. Nevertheless, BBPSO-MC algorithms are better than other algorithms for both VLE-LS and VLE-EIV problems at SC_{max} = 6D and 12D. Further, GSR results do not change significantly when the SC_{max} is increased from 6D to 12D for VLE-EIV problems (Fig. 7(b)). This suggests that for high dimensional VLE-EIV problems, use of SC-2 with SC_{max} = 6D is sufficient to obtain competitive results at relatively low computational resources.

6. Conclusions

With the introduction of mutation and crossover strategies for updating the *nbest*-index particle in the proposed BBPSO-MC algorithm, an adaptive balance between exploration and exploitation nature of search for the *nbest*-index particle is achieved. This discourages premature convergence of the search to local minima and improves the success rate. Different neighborhood topologies have been tested for the BBPSO-MC, and it has been concluded that BBPSO-MC using *lbest* with 2 neighbors outperforms the other topologies. The proposed BBPSO-MC-*lbest*-2N is shown to perform better than the original bare-bones algorithms on benchmark functions.

The PSO and variants of BBPSO algorithms have been applied for solving parameter estimation in VLE modeling problems based on LS and EIV approaches. Among the algorithms tested, BBPSO-MC was found to be the overall best performer across different VLE problems tried. In addition, comparison between BBPSO-MC and other stochastic algorithms (SA, DE, DETL, and PSO) suggests that the former provides higher success rate and efficiency for VLE modeling problems. Between the stopping criteria: SC-1 and SC-2, the former gave better results at the expense of computational resources for both VLE-LS and VLE-EIV problems. Furthermore, an increase of SC_{max} from 6D to 12D in SC-2 does not affect the results significantly for VLE-EIV problems. Thus the use of SC_{max} = 6D seems to be sufficient for VLE-EIV problems.

References

[1] P. Englezos, N. Kalogerakis, Applied Parameter Estimation for Chemical Engineers, Marcel-Dekker, New York, 2001.
[2] C.Y. Gau, M.A. Stadtherr, in: R. Luus (Ed.), Recent Developments in Optimization and Optimal Control in Chemical Engineering, Transword Research, 2002, pp. 1–17.
[3] C.Y. Gau, J.F. Brennecke, M.A. Stadtherr, Fluid Phase Equilibr. 168 (2000) 1–18.
[4] V.H. Alvarez, R. Larico, Y. Ianos, M. Aznar, Braz. J. Chem. Eng 25 (2008) 409–418.
[5] A. Bonilla-Petriciolet, J.G. Segovia-Hernández, Fluid Phase Equilibr. 289 (2010) 110–121.
[6] A. Bonilla-Petriciolet, G.P. Rangaiah, J.G. Segovia-Hernández, Fluid Phase Equilibria 287 (2010) 111–125.
[7] R.C. Eberhart, J. Kennedy, Proceedings of the Sixth International Symposium Micromachine Human Science, Nagoya, Japan, 1995, pp. 39–43.
[8] K.E. Parsopoulos, M.N. Vrahatis, Lect. Notes Comput. Sci. 3449 (2005) 590–599.
[9] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, IEEE Trans. Evol. Comput. 10 (2006) 281–296.
[10] M. Clerc, J. Kennedy, IEEE Trans. Evol. Comput. 6 (2002) 58–73.
[11] S.T. Hsieh, T.Y. Sun, C.C. Liu, S.J. Tsai, IEEE Trans. Cybern. 39 (2009) 444–456.
[12] G.H.M. Omran, A.P. Engelbrecht, A. Salman, Eur. J. Oper. Res. 196 (2008) 128–139.
[13] R.A. Krohling, E. Mendel, IEEE Congr. (2009) 3285–3291.
[14] R.C. Eberhart, Y. Shi, IEEE Congr.Comput. Intell. (1998) 69–73.
[15] J. Kennedy, Proceedings of the IEEE Swarm Intelligence Symposium, 2003, pp. 80–87.
[16] F. van den Bergh, A. Engelbrecht, Inform. Sci. 176 (2006) 937–971.
[17] A. Bonilla-Petriciolet, U.I. Bravo-Sanchez, F. Castillo-Borja, J.G. Zapiain-Salinas, J.J. Soto-Bernal, Braz. J. Chem. Eng. 24 (2007) 151–162.
[18] G.M. Bollas, P.I. Barton, A. Mitsos, Chem. Eng. Sci. 64 (2009) 1768–1783.
[19] M. Srinivas, G.P. Rangaiah, Comp. Chem. Eng. 30 (2006) 1400–1415.
[20] J.A. Lopez, V.M. Trejos, C.A. Cardona, Fluid Phase Equilibr. 248 (2006) 147–157.