Periodica Polytechnica
Chemical Engineering

59(3), pp. 186-200, 2015
DOI: 10.3311/PPch.7636
Creative Commons Attribution

RESEARCH ARTICLE

On the Performance of Swarm Intelligence Optimization Algorithms for Phase Stability and Liquid-Liquid and Vapor-Liquid Equilibrium Calculations

Seif-Eddeen K. Fateen[1,2*], Adrián Bonilla-Petriciolet[3]

Received 26 July 2014; accepted after revision 22 September 2014

Abstract

This study introduces new soft computing optimization techniques for performing the phase stability analysis and phase equilibrium calculations in both reactive and non-reactive systems. In particular, the performance of the several swarm intelligence optimization methods is compared and discussed based on both reliability and computational efficiency using practical stopping criteria for these applied thermodynamic calculations. These algorithms are: Intelligent Firefly Algorithm (IFA), Cuckoo Search (CS), Artificial Bee Algorithm (ABC) and Bat Algorithm (BA). It is important to note that no attempts have been reported in the literature to evaluate their performance in solving the phase and chemical equilibrium problems. Results indicated that CS was found to be the most reliable technique across different problems tried at the time that it requires similar computational effort to the other methods. In summary, this study provides new results and insights about the capabilities and limitations of bio-inspired optimization methods for performing applied thermodynamic calculations.

Keywords

Swarm intelligence, optimization methods, phase equilibrium, phase stability, chemical equilibrium

1 Introduction

Soft computing techniques are popular and reliable numerical tools to solve real world optimization problems especially those involved in engineering applications. In particular, nature-inspired algorithms are a branch in the field of soft computing, which imitate processes in nature/inspired from nature. Nature-inspired computation can be classified into six categories [1]: swarm intelligence, natural evolution, biological neural network, molecular biology, immune system and biological cells. To date, several nature-inspired algorithms have been developed for solving difficult non-convex and multivariable optimization problems. In particular, the sophisticated decision making process that swarms of living organisms exhibit has inspired several of these meta-heuristics. Examples of these swarm intelligence optimization techniques are based on the decision making process of fireflies, ants, bees or birds. In general, the bio-inspired methods are quite simple to implement and use. They do not require any assumptions or transformation of the original optimization problems, do not require good starting points, can easily move out of local minima in their path to the global minimum, and can be applied with any model (i.e., black box model), yet provide a high probabilistic convergence to the global optimum. They can often locate the global optimum in modest computational time compared to deterministic optimization methods [2]. Therefore, these techniques are more advantageous compared to traditional local gradient-based and global deterministic optimization techniques.

Recently, swarm intelligence optimization methods have been introduced for solving challenging global optimization problems involved in the thermodynamic modeling of phase equilibrium for chemical engineering applications [3-8]. In particular, the calculations of phase and chemical equilibrium are an essential component of all process simulators in chemical engineering. The prediction of phase behavior of a mixture involves the solution of two main thermodynamic problems: phase stability (PS) and phase equilibrium calculations (PEC). PS problems involve the determination of whether a system will remain in one phase at the given conditions or split into two or more phases. This type of problems usually precedes

[1] Department of Chemical Engineering, Cairo University, Giza, Egypt

[2] Department of Petroleum and Energy Engineering, American University in Cairo, New Cairo, Egypt

[3] Department of Chemical Engineering, Instituto Tecnológico de Aguascalientes, México

[*] Corresponding author, e-mail: fateen@eng1.cu.edu.eg

the PEC problem, which involves the determination of the number, type and composition of the phases at equilibrium at the given operating conditions. Note that a reactive phase equilibrium calculation (RPEC) or chemical equilibrium calculation is performed if any reaction is possible in the system under study. During the analysis of a chemical engineering process, PS, PEC and/or RPEC problems usually need to be solved numerous times. Solving these types of thermodynamic problems involves the use of global optimization methods. In particular, PS analysis requires the minimization of the tangent plane distance function (TPDF), while the Gibbs free energy function needs to be minimized for PEC and RPEC subject to the corresponding constraints [9]. For these thermodynamic problems, finding a local minimum is not sufficient; and the global minimum must be identified for determining the correct thermodynamic condition.

In general, the high non-linearity of thermodynamic models, the non-convexity of the objective functions, and the presence of a trivial solution in the search space make PEC, RPEC and PS problems difficult to solve. Moreover, these thermodynamic problems may have local optimal values that are very comparable to the global optimum value, which makes it challenging to find the global optimum. Hence, PS, PEC, and RPEC problems require a reliable and efficient global optimization algorithm. To date, there are no effective optimization methods at all for performing these thermodynamic calculations. Current methods for phase equilibrium modeling have their own deficiencies and sometime fail to find the correct solutions for difficult problems such as the calculation of simultaneous phase and chemical equilibrium for systems containing many components near the critical point of the mixture and the phase boundaries [10]. Actually, novel processes in chemical industry handle complex mixtures, severe operating conditions, or even incorporate combined unit operations (e.g., reactive distillation or extractive distillation). Wrong estimation of the thermodynamic state may have negative impacts on the design, analysis and operation of such novel processes. Therefore, the search for better methods and techniques to solve these often-difficult thermodynamic problems is still ongoing and new optimization algorithms should be developed and/or analyzed.

Swarm intelligence optimization methods have been insufficiently studied in chemical engineering applications including the thermodynamic modeling of phase equilibrium. In particular, this work evaluates a set of promising bio-inspired optimization algorithms for PEC, RPEC and PS problems involving multiple components, multiple phases and popular thermodynamic models. These algorithms are: Intelligent Firefly Algorithm (IFA), Cuckoo Search (CS), Artificial Bee Algorithm (ABC) and Bat Algorithm (BA). The performance of these swarm intelligence stochastic global optimization algorithms has not been comparatively studied before for phase stability and equilibrium problems. In this work, they compared and discussed based on both reliability and computational efficiency using practical stopping criteria. In summary, this study provides new results and insights about the capabilities and limitations of bio-inspired optimization methods for performing applied thermodynamic calculations. The remainder of this paper is organized as follows. The four algorithms, (i.e., IFA, CS, ABC, BA) are presented in Section 2. A brief description of PEC, PS and RPEC problems is given in Section 3. Implementation of the four algorithms is covered in Section 4 and Section 5 presents the results and discusses the performance of the bio-inspired methods on selected thermodynamic problems. Finally, the conclusions of this work are summarized in Section 6.

2 Description of Bio-Inspired Optimization Algorithms used for thermodynamic calculations

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

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

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. This optimization problem can be subject to constraints depending on the type of thermodynamic calculation (i.e., PS, PEC or RPEC). These constraints are presented with the thermodynamic problems in the following section.

Four different stochastic global optimization techniques, Intelligent Firefly Algorithm (IFA), Cuckoo Search (CS), Artificial Bee Algorithm (ABC) and Bat Algorithm (BA), were evaluated for the phase stability and equilibrium calculations in this study. These methods were selected because they are relatively new with improved global optimization features. It is important to note that no attempts have been reported in the literature to evaluate their performance in solving the phase and chemical equilibrium problems and it is expected that their performance could be superior to other stochastic methods. Each of these methods is briefly described in the following subsections. More details of these stochastic optimization methods can be found in the cited references.

2.1 Intelligent Firefly Algorithm

Firefly Algorithm (FA) is a nature-inspired meta-heuristic stochastic global optimization method that was developed by Yang [11]. It is a relatively new method that is gaining popularity in finding the global minimum of diverse applications. It was rigorously evaluated by Gandomi et al. [12], and has been recently used to solve the flow shop scheduling problem [13], financial portfolio optimization [14], and phase and chemical equilibrium problems [5]. The FA algorithm imitates the mechanism of firefly communications via luminescent flashes. In the FA algorithm, the two important issues are the variation of light intensity and the formulation of attractiveness. The brightness of a firefly is determined by the landscape of the objective function. Attractiveness is proportional to brightness and, thus, for any two flashing fireflies, the less bright one moves towards the brighter one.

In this algorithm, the attractiveness of a firefly is determined by its brightness, which is equal to the objective function. The brightness of a firefly at a particular location x was chosen as $I(\boldsymbol{x}) = f(\boldsymbol{x})$. The attractiveness is judged by the other fireflies. Thus, it was made to vary with the distance between firefly i and firefly j. The attractiveness was made to vary with the degree of absorption of light in the medium between the two fireflies. Thus, the attractiveness is given by

$$\beta = \beta_{min} + \left(\beta_o - \beta_{min}\right) e^{-\gamma r^2} \tag{2}$$

The distance between any two fireflies i and j at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is the Cartesian distance:

$$r_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2} \tag{3}$$

The movement of a firefly attracted to another more attractive (brighter) firefly j is determined by

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \beta\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right) + \alpha\epsilon_i \tag{4}$$

The second term is due to the attraction, while the third term ε_i is a vector of random numbers drawn from a uniform distribution in the range [-0.5, 0.5].

In the original FA, the move Eq. (4) is determined mainly by the attractiveness of the other fireflies; the attractiveness is a strong function of the inter distance between the fireflies. Thus, a firefly can be attracted to another firefly merely because it is close, which may take it away from the global minimum. The fireflies are ranked according to their brightness, i.e. according to the values of the objective function at their respective locations. However, this ranking, which is a valuable piece of information per se, is not utilized in the move equation. A firefly is pulled towards each other firefly as each of them contributes to the move by its attractiveness. This behavior may lead to a delay in the collective move towards the global minimum. The idea behind of Intelligent Firefly Algorithm (IFA) is to make use of the ranking information such that every firefly is moved by the attractiveness of a fraction of fireflies only and not by all of them [4]. This fraction represents a top portion of the fireflies based on their rank. Thus, a firefly is acting intelligently by basing its move on the top ranking fireflies only and not merely on attractiveness.

```
Step 1: Set parameters and Initialize
    Generate initial population of n fireflies $x_i$
    Light intensity $I_i$ at $x_i$ is determined by $f(x_i)$
    Define light absorption coefficient $\gamma$
    Define fraction of top fireflies $\varphi$

Step 2: Main evolution loop
while (t < MaxGeneration)
        for i=1:n
                for j=1:n
                        if ($I_i < I_j$) & j belongs to top fireflies, Move
                        firefly i towards j; end if
                        Vary attractiveness with distance r via exp[-$\gamma$r]
                        Evaluate new solutions and update light intensity
                end for j
        end for i
        Rank the fireflies and find the current best solution.
end while

Local optimization starting from the best solution found by the global search
Output: Solution found by local optimizer.
```

Fig. 1 Simplified algorithm of IFA

A simplified algorithm for the IFA technique is presented in Fig. 1. The new parameter φ is the fraction of the fireflies utilized in the determination of the move. The original firefly algorithm is retained by setting φ to 1. This parameter is used as the upper limit for the index j in the inner loop. Thus, each firefly is moved by the top φ fraction of the fireflies only. The strength of FA is that the location of the best firefly does not influence the direction of the search. Thus, the fireflies are not trapped in a local minimum. However, the search for the global minimum requires additional computational effort as many fireflies wander around uninteresting areas. With the intelligent firefly modifications, the right value of the parameter φ can maintain the advantage of not being trapped in a local minimum while speeding up the search for the global minimum. The parameters for the original FA were kept constant in all experiments for IFA: we used the value of 1 for β_o, 0.2 for β_{min}, 1 for γ, and α was made to decrease with the increase in the iteration number, k, in order to reduce the randomness according to the following formula

$$\alpha_k = \left(1.11\times10^{-4}\right)^{b/iter_{max}} \alpha_{k-1} \tag{5}$$

Thus, the randomness is decreased gradually as the optima are approached. This formula was adapted from Yang [10]. The value of the parameter b was taken equal to 5.

2.2 Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) is a meta-heuristic optimization algorithm based on the foraging behavior of bee swarms [15]. In the ABC algorithm, the colony of artificial bees contains three groups: employed bees (typically 50% of initial population), onlookers (typically 50% of initial population) and scouts (typically one scout bee). The quality of a food source is the amount of nectar it retains as a metaphor for the value of the objective function. For every food source, there is only one employed bee. The search carried out by the bees can be summarized as follows: (1) employed bees determine a food source within the neighborhood of a food source in their memory; (2) employed bees share their information with onlookers, and then onlookers select one of the food sources; (3) onlookers select a food source within the neighborhood of the food sources chosen by themselves, thus they perform probabilistically guided exploitation, and (4) an employed bee retains its right to convert to a scout by abandoning a food source after its exemption, as judged by the failure of a prescribed number of inner iterations (parameter *limit)* to improve the abandoned food source. In step 1, the employed bees search their neighborhood as guided by the following formula

$$v_{ij} = x_{ij} + \Phi_{ij}\left(x_{ij} - x_{kj}\right) \tag{6}$$

where v_{ij} is the position of the new food source in the neighborhood of x_{ij}, with k as the solution, and Φ is a random number in the range [-1, 1]. The onlookers, in step 3, apply a greedy selection criterion that is proportional to the fitness value of the searched sources. In step 4, sources abandoned are randomly replaced by the following formula

$$x_{ij} = min_j + \varphi_{ij}\left(max_j - min_j\right) \tag{7}$$

where φ_{ij} is a random number in the range [0,1]. Pseudo code of ABC algorithm is given in Fig. 2. ABC algorithm, its variants, and its hybrids have shown success in a number of applications from electrical, mechanical, civil, electronics, software and control engineering. Interested readers can consult [16] for a comprehensive survey on ABC and its applications.

```
Step 1: Set parameters and Initialize
    Generate food sources $x_i$
    Evaluate fitness of the population $f(x_i)$
    Define parameter limit

Step 2: Main loop
while (t < MaxGeneration)
        for i=1:n/2 {Employed phase}
        Evaluate solutions $v_{ij}$ and $x_{ij}$
        if ($f(v_{ij}) < f(x_{ij})$), Do greedy selection; else $count_i = count_i + 1$; end if
        end for
        for i=n/2+1 : n {Onlooker phase}
        Calculate selection probability
        Produce a new solution $v_{ij}$ from the selected bee
        Evaluate solutions $v_{ij}$ and $x_{ij}$
        if ($f(v_{ij}) < f(x_{ij})$), Do greedy selection; else $count_i = count_i + 1$; end if
        end for
        for i=1:n {Scout phase}
        if ($count_i > limit$), $x_{ij}$ = random; end if
        end for
        Rank the food sources and find the current best solution.
end while

Local optimization starting from the best solution found by the global search
Output: Solution found by local optimizer.
```

Fig. 2 Simplified algorithm of ABC

2.3 Cuckoo Search Algorithm

Cuckoo Search (CS) is a meta-heuristic optimization algorithm based on the breeding behavior of certain species of cuckoos. Some species lay their eggs in the nest of other host birds. If a host bird discovers the alien eggs, they will either throw them away or abandon the nest and build a new one elsewhere. As explained by Yang and Deb [16], the CS is based on three rules: (1) each cuckoo lays one egg at a time and dump its egg in a randomly chosen nest; (2) the best nest with high quality of eggs will carry over to the next generations; and (3) the number of available host nests is fixed, and a fraction p_a of the n nests are replaced by new nests (with new random solutions). Each egg represents a new solution and each nest can hold a single egg only. The aim is to use the new and potentially better solutions to replace the worse solutions in the nests. When generating new solutions for Cuckoo i, a Lévy flight is performed. A stochastic equation for random walk is used to represent the search of each cuckoo. The next location of a cuckoo depends on its current location and the transition probability. The step length is randomly drawn for a Lévy distribution, which has an infinite variance with an infinite mean. Some of the new solutions should be generated by Lévy walk around the best solution obtained so far, which will speed up the local search. Some other solutions should be generated by far field randomization to avoid being trapped in a local optimum. The pseudo code of CS is depicted in Fig. 3.

```
Step 1: Set parameters and Initialize
    Generate n host nests $x_i$
    Define fraction $p_a$

Step 2: Main loop
while (t < MaxGeneration)
        Get a cuckoo randomly by Lévy flight
        Evaluate its fitness $F_i$
        Choose randomly a nest j among n
        If ($F_i > F_j$), Do greedy selection; end if
        Abandon a fraction $p_a$ of worse nests for the sake of randomly
        generated ones
        Keep best solutions
        Rank solutions and find the current best solution.
end while

Local optimization starting from the best solution found by the global search
Output: Solution found by local optimizer.
```

Fig. 3 Simplified algorithm of CS

2.4 Bat Algorithm

Bats have advanced capability of echolocation, which is a type of sonar used by bats to detect prey and avoid obstacles. They emit a loud sound pulse and listen for the echo that bounces back from the surrounding objects. Some types of bats use the time delay from the emission and detection of the echo, the time difference between their two ears, and the loudness variations of the echoes to build up three-dimensional scenario of the surrounding [17]. They can detect the distance and orientation of the target, the type of prey, and even the moving speed of the prey such as small insects. Indeed, they seem to be able to discriminate targets by the variations of the Doppler Effect induced by the wing-flutter rates of the target insects. Such echolocation behavior of bats was formulated in such a way that it can be associated with the objective function to be optimized [18]. In short, BA uses the following approximate or idealized rules:

1. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way;
2. Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target;
3. The loudness is assumed to vary from a large (positive) A_0 to a minimum constant value A_{min}.

The frequency f in a range $[f_{min}, f_{max}]$ corresponds to a range of wavelengths $[\lambda_{min}, \lambda_{max}]$. In the actual implementation, the detectable range (or the largest wavelength) should be chosen such that it is comparable to the size of the domain of interest, and then toned down to smaller ranges. BA algorithm has recently demonstrated its ability to solve tough optimization problems, like continuous optimization for engineering design, combinatorial optimization and scheduling, parameter estimation, image processing and data mining [20]. The pseudocode shown in Fig. 4 shows the basic steps of the Bat Algorithm.

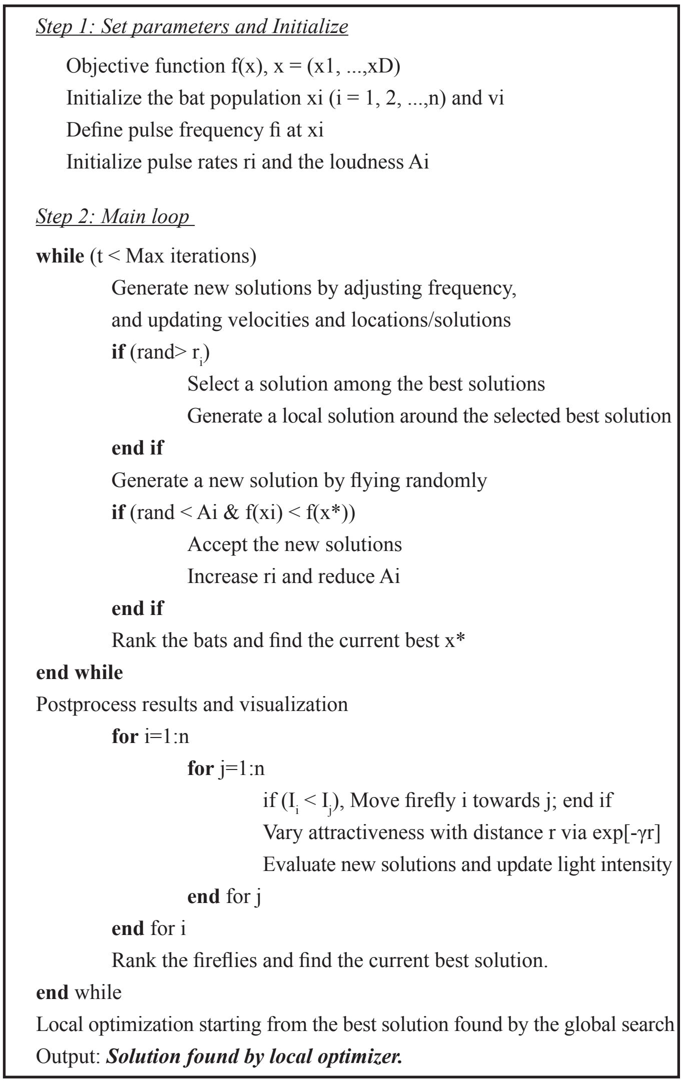
3 Formulation of thermodynamic calculations for phase equilibrium modeling: PS, PEC and RPEC Problems

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 subsections. It is convenient to remark that we have used the nomenclature and mathematical formulations reported in [9] for describing these thermodynamic problems.

3.1 Phase Stability Analysis Problems

Solving the PS problem is usually the starting point for the phase equilibrium calculations. The theory used to solve this problem states that a phase is stable if the tangent plane generated at the feed (or initial) composition lies below the molar Gibbs energy surface for all compositions. One common implementation of the tangent plane criterion 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 [21]. TPDF is given by

$$TPDF = \sum_{i=1}^{c} y_i (\mu_i|_y - \mu_i|_z) \tag{8}$$



```
Step 1: Set parameters and Initialize
    Objective function f(x), x = (x1, ...,xD)
    Initialize the bat population xi (i = 1, 2, ...,n) and vi
    Define pulse frequency fi at xi
    Initialize pulse rates ri and the loudness Ai

Step 2: Main loop
while (t < Max iterations)
        Generate new solutions by adjusting frequency,
        and updating velocities and locations/solutions
        if (rand> r_i)
                Select a solution among the best solutions
                Generate a local solution around the selected best solution
        end if
        Generate a new solution by flying randomly
        if (rand < Ai & f(xi) < f(x*))
                Accept the new solutions
                Increase ri and reduce Ai
        end if
        Rank the bats and find the current best x*
end while
Postprocess results and visualization
        for i=1:n
                for j=1:n
                        if (I_i < I_j), Move firefly i towards j; end if
                        Vary attractiveness with distance r via exp[-γr]
                        Evaluate new solutions and update light intensity
                end for j
        end for i
        Rank the fireflies and find the current best solution.
end while
Local optimization starting from the best solution found by the global search
Output: Solution found by local optimizer.
```

Fig. 4 Simplified algorithm of BA

where $\mu_i|_y$ and $\mu_i|_z$ are the chemical potentials of component i calculated at compositions y and z, respectively. 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 negative, the phase is not stable at the given conditions, and phase split calculations are necessary to identify the compositions of each phase. The decision variables for minimizing $TPDF$ in phase stability problems are mole fractions, y_i for $i = 1, 2, \ldots, c$, each in the range [0, 1], and the constraint is that summation of these mole fractions is equal to 1. According to [9], 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

$$n_{iy} = \beta_i z_i n_F \quad i = 1, \ldots, c \tag{9}$$

and

$$y_i = \frac{n_{iy}}{\sum_{j=1}^{c} n_{jy}} \quad i = 1, \ldots, c \tag{10}$$

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 unconstrained global optimization problem for phase stability analysis becomes [9]

$$\begin{gathered} \min TPDF\left(\beta_i\right) \\ 0 \le \beta_i \le 1,\ i = 1 \ldots c \end{gathered} \tag{11}$$

More details on PS problem formulation can be found in [19] and the characteristics of PS problems used in this study are summarized in Table 1.

3.2 Phase Equilibrium Calculation 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 differ significantly in different phases at equilibrium. If there is no reaction between the different substances, then it is a phase equilibrium problem. Classic thermodynamics indicate that minimization of Gibbs free energy is a natural approach for calculating the equilibrium state of a mixture [9]. The mathematical formulation involves the minimization of Gibbs free energy subject to mass balance equality constraints and bounds that limit the range of decision variables. In a non-reactive system with c components and π phases, the objective function for PEC is

$$g = \sum_{j=1}^{\pi} \sum_{i=1}^{c} n_{ij} \ln(x_{ij}\gamma_{ij}) = \sum_{j=1}^{\pi} \sum_{i=1}^{c} n_{ij} \ln\left(\frac{x_{ij}\hat{\phi}_{ij}}{\phi_i}\right) \tag{12}$$

where n_{ij}, x_{ij}, γ_{ij}, $\hat{\phi}_{ij}$ and ϕ_i are the moles, mole fraction, activity coefficient and fugacity coefficient of component i in phase j, and the fugacity coefficient of pure component, respectively. Thermodynamic function (12) must be minimized with respect to n_{ij} taking into account the following mass balance constraints:

$$\sum_{j=1}^{\pi} n_{ij} = z_i n_F \quad i = 1,\ldots,c \tag{13}$$

$$0 \le n_{ij} \le z_i n_F \quad i = 1,\ldots,c \quad j = 1,\ldots,\pi \tag{14}$$

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, one can use new variables instead of n_{ij} as decision variables [9]. For multi-phase non-reactive systems, new variables β_{ij} ϵ (0, 1) are defined and employed as decision variables by using the following expressions

$$n_{i1} = \beta_{i1} z_i n_F \quad i = 1,\ldots,c \tag{15}$$

$$n_{ij} = \beta_{ij}\left(z_i n_F - \sum_{m=1}^{j-1} n_{im}\right) \quad i = 1,\ldots,c; \quad j = 2,\ldots,\pi - 1 \tag{16}$$

$$n_{i\pi} = z_i n_F - \sum_{m=1}^{\pi-1} n_{im} \quad i = 1,\ldots,c \tag{17}$$

For Gibbs energy minimization, the number of decision variables β_{ij} is c (π - 1) for non-reactive systems. The details of PEC problems used in this study are also in Table 1. In most of the reported studies, PEC problems tested assumed 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.

3.3 Reactive Phase Equilibrium Calculation 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 with respect to mole numbers of components in each phase subject to element/mass balances and chemical equilibrium constraints [9]. In this study, we have used a constrained Gibbs free energy minimization approach for performing RPEC. Specifically, for a system with c components and π phases subject to r independent chemical reactions, the objective function for RPEC is defined as [3,9]

$$F_{obj} = g \text{ - } \sum_{j=1}^{\pi} \ln \mathbf{K}_{eq} \mathbf{N}^{-1} \mathbf{n}_{ref,j} \tag{18}$$

where g is given by Eq. (12), ln $\mathbf{K}_{eq}$ is a row vector of logarithms of chemical equilibrium constants for r independent reactions, $\mathbf{N}$ is an invertible, square matrix formed from the stoichiometric coefficients of a set of reference components chosen from r reactions, and $\mathbf{n}_{ref}$ 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]

$$\begin{gathered} \sum_{j=1}^{\pi}\left(n_{ij} - \mathbf{v}_i \mathbf{N}^{-1} \mathbf{n}_{ref,j}\right) = n_{iF} - \mathbf{v}_i \mathbf{N}^{-1} \mathbf{n}_{ref,F} \\ i = 1,\ldots c - r \end{gathered} \tag{19}$$

where $n_{i,F}$ is the initial moles of component i in the feed. These mass balance equations are rearranged to reduce the number of decision variables in the optimization problem and to eliminate equality constraints, then

$$\begin{gathered} n_{i\pi} = n_{iF} - \mathbf{v}_i \mathbf{N}^{-1}(\mathbf{n}_{ref,F} - \mathbf{n}_{ref,\pi}) - \sum_{j=1}^{\pi-1}(n_{ij} - \mathbf{v}_i \mathbf{N}^{-1} \mathbf{n}_{ref,j}) \\ i = 1,\ldots,c - r \end{gathered} \tag{20}$$

Table 1 Details of PEC and PS problems studied

PEC/PS No.	System	Feed conditions	Thermodynamic models [1]	Global optimum for	
				Equilibrium	Stability
1	n-Butyl acetate + water	n_F = (0.5, 0.5) at 298K and 101.325kPa	NRTL model	-0.020198	-0.032466
2	Toluene + water + aniline	n_F = (0.29989, 0.20006, 0.50005) at 298K and 101.325kPa	NRTL model	-0.352957	-0.294540
3	$N_2+C_1+C_2$	n_F = (0.3, 0.1, 0.6) at 270K and 7600kPa	SRK EoS with classical mixing rules	-0.547791	-0.015767
4	$C_1 + H_2S$	n_F = (0.9813, 0.0187) at 190K and 4053kPa	SRK EoS with classical mixing rules	-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 390K and 5583kPa	SRK EoS with classical mixing rules	-1.183653	-0.000002
6	$C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_{7\text{-}16} + C_{17+}$	n_F = (0.7212, 0.09205, 0.04455, 0.03123, 0.01273, 0.01361, 0.07215, 0.01248) at 353K and 38500kPa	SRK EoS with classical mixing rules	-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 314K and 2010.288kPa	SRK EoS with classical mixing rules	-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 435.35K and 19150kPa	SRK EoS with classical mixing rules	-1.121176	-0.0000205

[1] Details of thermodynamic models are reported in references [3-6].

Using Eq. (20), 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. (18) with these decision variables and the remaining $c - r$ mole numbers ($n_{i\pi}$) are determined from Eq. (20), subject to the inequality constraints $n_{i\pi} > 0$. The penalty function method is used to solve the constrained Gibbs free energy minimization in reactive systems and the optimization problem is defined as [9]

$$F_r = \begin{cases} F_{obj} & \text{if } \forall n_{ij} > 0 \quad i = 1,\ldots,c \quad j = 1,\ldots,\pi, \\ F_{obj} + p & \text{otherwise,} \end{cases} \tag{21}$$

where p is the penalty term whose value is positive and given by

$$p = 10 \cdot \sum_{i=1}^{n_{unf}} |n_{i\pi}| \tag{22}$$

where $n_{i\pi}$ is obtained from Eq. (20) and n_{unf} is the number of infeasible mole numbers (i.e., $n_{i\pi} < 0$ where $i = 1, \ldots, c - r$). The details of the RPEC problems are shown in Table 2.

4 Implementation of the Bio-Inspired Optimization Algorithms

In this study, all the bio-inspired optimization algorithms and thermodynamic models were coded in MATLAB®. The parameters used for the bio-inspired algorithms were tuned for each type of thermodynamic calculation and then they were fixed for all problems tested in order to compare the robustness of the algorithms, see Table 3 for their values. Further, $NP = 10D$ was used for all the optimization methods. Altogether, we studied 24 problems consisting of 8 PEC, 8 PS and 8 RPEC problems. All these problems are multimodal with number of decision variables ranging from 2 to 10. Each thermodynamic problem was solved 100 times independently with a different random number seed for robust performance analysis. The performances of bio-inspired algorithms were compared based on Success Rate (SR) and average number of function evaluations (for both global and local searches) in all 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 best objective function value (SC_{max}). These stopping conditions have been employed in previous studies on bio-inspired computation for phase equilibrium modeling [4-8] and correspond to common convergence criteria for solving real-world optimization problems. 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. On the other hand, SR is the number of times the algorithm located the global optimum to the specified accuracy,

Table 2 Details of RPEC problems studied

RPEC No.	System	Feed conditions	Thermodynamic models [1]	Global optimum
1	A1+A2 ↔ A3+A4 (1) Ethanol (2) Acetic acid (3) Ethyl acetate (4) Water	n_F = (0.5, 0.5, 0.0, 0.0) at 355K and 101.325kPa	NRTL model and ideal gas. K_{eq}=18.670951	-2.05812
2	A1+A2 ↔ 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 373.15K and 101.325kPa	Wilson model and ideal gas. $\Delta G^0_{rxs}/R = -4205.05 +10.0982T-0.2667T\ln T$ $\ln K_{eq} = -\Delta G^0_{rxs}/R$ where T is in K	-1.434267
3	A1+A2 +2A3↔ 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 355K and 151.95kPa	Wilson model and ideal gas. $K_{eq}=1.057*10^{-04}e^{4273.5/T}$ where T is in K	-1.226367
4	A1+A2 ↔ 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 298.15K and 101.325kPa	UNIQUAC model and ideal gas. $\ln K_{eq}=450/T +0.8$	-1.10630
5	A1+A2 ↔ A3	n_F = (0.6, 0.4, 0.0)	Margules solution model. $g^E/R_gT = 3.6x_1x_2+2.4x_1x_3+2.3x_2x_3$ K_{eq}=0.9825	-0.144508
6	A1+A2+2A3 ↔ 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 335K and 151.9875kPa	Wilson model and ideal gas. $K_{eq}=1.057*10^{-04}e^{4273.5/T}$ where T is in K	-0.872577
7	A1+A2 ↔ A3	n_F = (0.52, 0.48, 0.0) at 323.15K and 101.325kPa	Margules solution model. K_{eq} = 3.5	-0.653756
8	A1+A2 ↔ A3+A4	n_F = (0.048, 0.5, 0.452, 0.0) at 360K and 101.325kPa	NRTL model K_{eq}=4.0	-0.311918

out of 100 runs. A run/trial is considered successful if the best 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 and is defined as

$$GSR = \sum_{i=1}^{np} \frac{SR_i}{np} \tag{23}$$

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

At the end of each run by each stochastic algorithm, a local optimizer was 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 performance analysis; however, global search in the subsequent iterations is not affected by this. Since all algorithms were implemented in MATLAB®, sequential quadratic program (SQP) was chosen as the local optimizer. The best solution at the end of the stochastic algorithm was 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. In the small number of cases when the local optimizer diverged to a larger value of the objective function, the output of the stochastic algorithm was retained. All computations were performed on 64-bit HP Pavilion dv6 Notebook computer with an Intel Core i7-2630QM processor, 2.00 GHZ and 4 GB of RAM, which can complete 1344 MFlops (million floating-point operations) for the LINPACK benchmark program that uses the MATLAB "backslash" operator to solve for a matrix of order 500.

Table 3 Selected values of the parameters used in the implementation of the bio-inspired algorithms

Method	Parameter	Selected value
ABC	*n*	10 D
	Foodnumber	n/2
	limit	100
CS	*n*	10 D
	p_a	0.25
FA	α_o	0.5
	β_{min}	0.2
	γ	1
	n	10 D
IFA	*Same as FA* + ϕ	0.05
BA	*n*	10 D
	A	0.25
	r	0.5

5 Results and Discussion

5.1 Performance of Bio-inspired Algorithms on PS problems

On PS problems, similar tests using the four bio-inspired algorithms in addition to FA were performed. Results were collected at different iteration levels, starting from 10-iteration level, after local optimization at each of these iteration levels. As expected, GSR of ABC, CS, FA, IFA, and BA for all PS problems using SC-1 improves with increasing number of iterations (Fig. 5a). The highest GSR was 99.25% % obtained by the CS algorithm without the local optimization. The selected PS problems were somewhat difficult to optimize, which is reflected in the relatively low GSR without using the local optimizer. At 10 and 25 iterations, BA obtained the best GSR, but from 50 to 750 iterations, the CS obtained the best GSR. At the termination of the iterations, CS obtained the highest GSR. The reliability of IFA and CS did not improve much beyond the 750th iteration. However, reliability of FA, BA and ABC kept improving until the end.

Figure 5b shows the improvements in GSR with the use of a local optimization technique at the end of the stochastic techniques. The best GSR, which was obtained by CS, did not increase whatsoever with the use of local optimization. The performance of CS showed slightly more reliability compared to that of ABC at higher iterations, while the opposite was true for lower iterations. Another interesting observation from Fig. 5b is that GSR of CS, FA, and IFA using local optimization decreased with iterations to reach a minimum, and then veered into an increase. Since PS problems are particularly difficult, local optimization techniques may diverge even with an improved initial point obtained by further iterations in the stochastic method. Only at higher iterations that the normal increase in GSR with increased iterations for the three algorithms was retained.

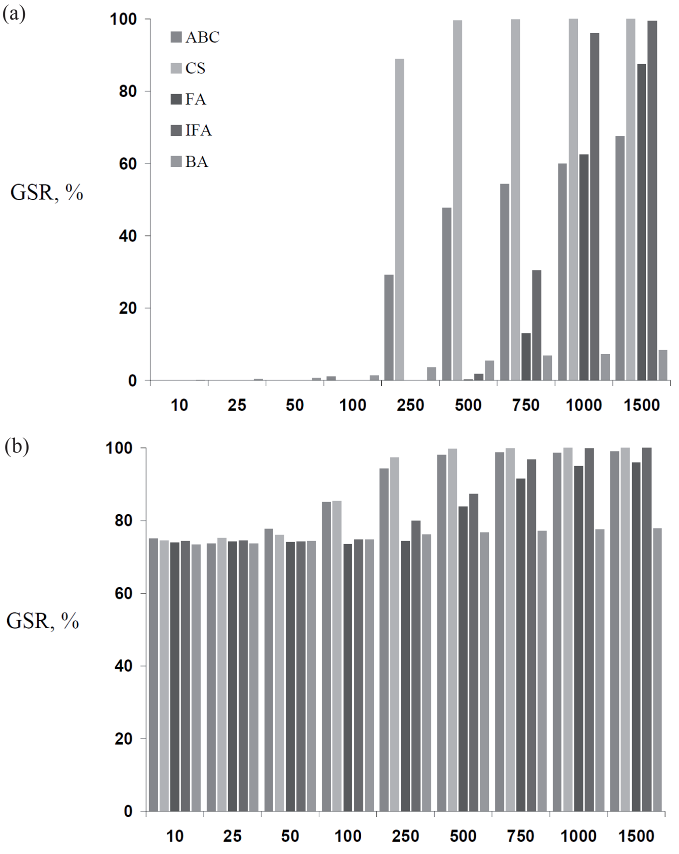


Fig. 5 Global Success Rate (GSR) versus Iterations for PS problems using ABC, CS, FA, IFA, and BA with SC-1: (a) bio-inspired method only and (b) bio-inspired method combined with local optimization

In stochastic global optimization, 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 ABC, CS, FA, IFA, and BA for all PS problems are presented in Fig. 6; GSR and NFE reported in this table are for stochastic methods followed by local optimization. They show that, in general, reliability of the algorithm and NFE increase with increasing SC_{max}. However, the reliability obtained using SC-1 is always higher than that obtained by SC-2, which is shown in Fig. 6a that summarizes GSR of the five techniques with the three SC-2 stopping criteria compared with SC-1 = 1500 for all PS problems. Fig. 6b shows the NFE of the five techniques with the four stopping criteria for all PS problems. FA, IFA, and BA were almost insensitive to the increase in SC-2, while ABC and CS showed improved reliability with increased SC_{max}. For all five methods, SC-1 needed much higher NFE than all SC-2 stopping criteria, reaching its zenith at 8 times more NFE. Hence, for bio-inspired algorithms that show comparable reliability for both stopping criteria; algorithms like ABC and CS, it is advisable to use SC-2, which offers much higher efficiency at the expense of only slightly diminished reliability.

Problems 6, 7 and 8 were identified as challenging since at least one of the methods failed to achieve 50% GSR even

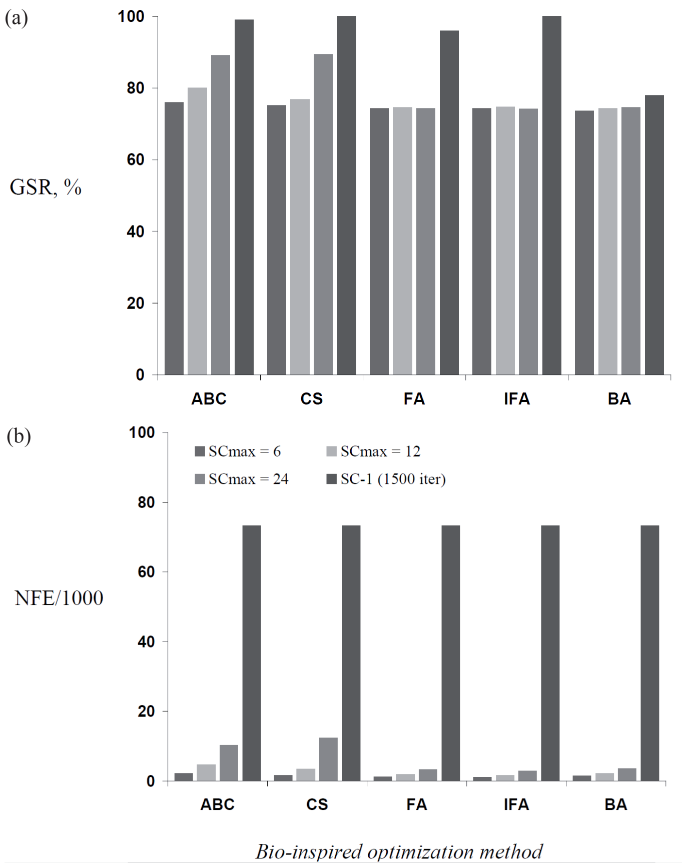


Fig. 6 (a) Global Success Rate (GSR) and (b) average NFE of ABC, CS, FA, IFA, and BA for PS problems using SC-2 (SC_{max} = 10, SC_{max} = 25 and SC_{max} = 50) and SC-1 (1500 iterations)

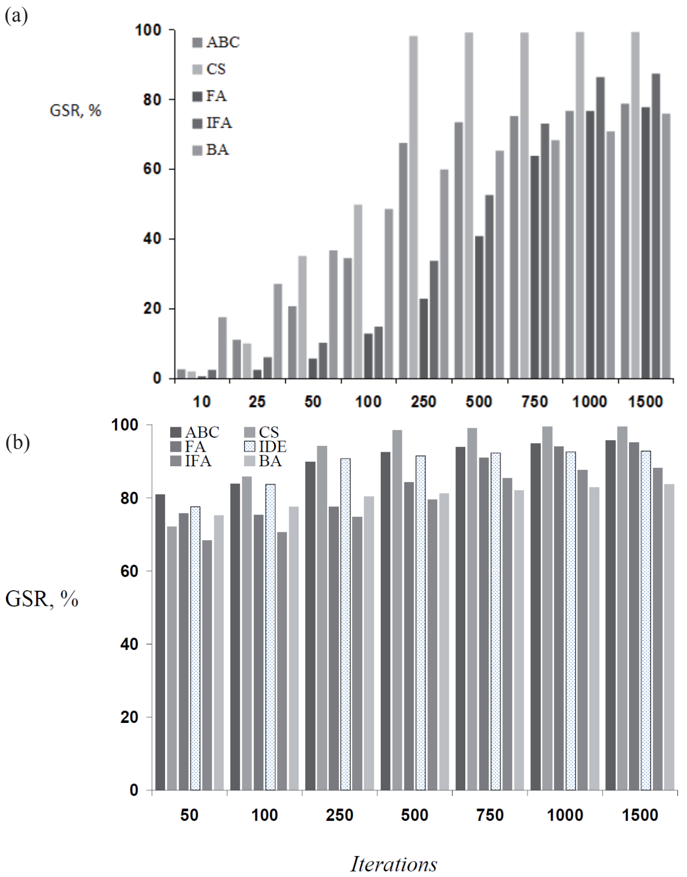


Fig. 7. Global Success Rate (GSR) versus Iterations for PEC problems using ABC, CS, FA, IFA, and BA with SC-1: (a) bio-inspired method only and (b) bio-inspired method combined with local optimization

with the subsequent local optimization. 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 [9]. So, better performance of stochastic methods is expected if more iterations and/or larger population size are used.

5.2 Performance of Bio-inspired Algorithms on PEC problems

GSR values for all PEC problems by ABC, CS, FA, IFA, and BA algorithms with NP of 10D using SC-1 are illustrated in Fig. 7. As expected, GSR improves with increasing number of iterations, particularly at lower iteration levels. After 250 iterations, GSR almost does not improve at all for CS. However, GSR kept improving for the rest of the bio-inspired methods. In general, subsequent iterations without improvement in the results are waste of computational resources. For example, for bio-inspired optimization only, GSR of CS is 98% at 250 iterations; it increases to 99.125% at 500 iterations and stays approximately the same until 1500 iterations. Results in Fig. 7a show that BA has higher reliability at low NFE compared to the rest of algorithms, for PEC problems when global stochastic optimization only is used. When the performance of the five methods with local optimization at the end of the global search is compared, CS gives the highest reliability with GSR close to 100% at 1500 iterations.

The effect of stopping criterion, SC-2 on ABC, CS, FA, IFA, and BA algorithms has also been studied on PEC problems. Figure 8 summarizes GSR and NFE of ABC, CS, FA, IFA, and BA algorithms with four stopping criteria. We obtained the same conclusion of higher reliability with higher SC_{max}. It can be observed in Fig. 8a that the use of SC-2 gives lower GSR when compared to that with SC-1 for all five methods. NFE values in Fig. 8b shows that CS uses most NFE to terminate the global search by SC-2 compared to the rest of the algorithms. In general, SC-2 requires significantly fewer NFE compared to SC-1, which confirms the need for a good termination criterion. Especially, with SC_{max} = 50, SR obtained by the ABC, CS, and BA is marginally smaller than that obtained with SC-1 but uses much fewer NFE, see Fig. 8. BA achieved comparable reliability to ABC, FA, and IFA with fewer NFE when SC-2 was used. When SC-1 was used, CS achieved better reliability with almost identical NFE as compared to the other four methods.

5.3 Performance of Bio-inspired Algorithms on RPEC problems

GSR of ABC, CS, FA, IFA, and BA algorithms for all RPEC problems using SC-1 is reported in Fig. 9a, when global stochastic optimization was used without the subsequent use of local optimization. GSR generally improves with increasing number of

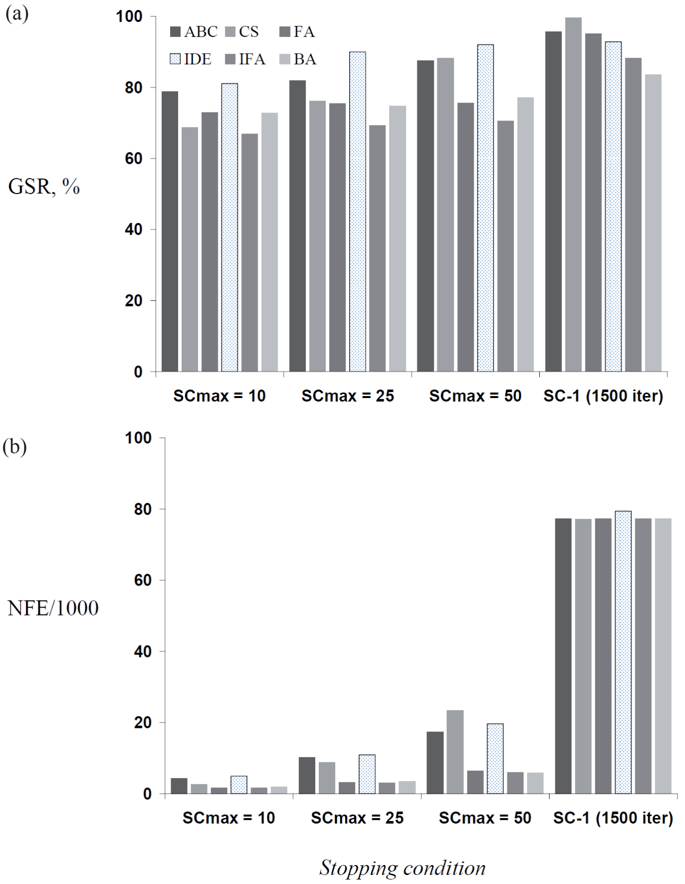


Fig. 8 (a) Global Success Rate (GSR) and (b) average NFE of ABC, CS, FA, IFA, and BA for PEC problems using SC-2 (SC_{max} = 10, SC_{max} = 25 and SC_{max} = 50) and SC-1 (1500 iterations)

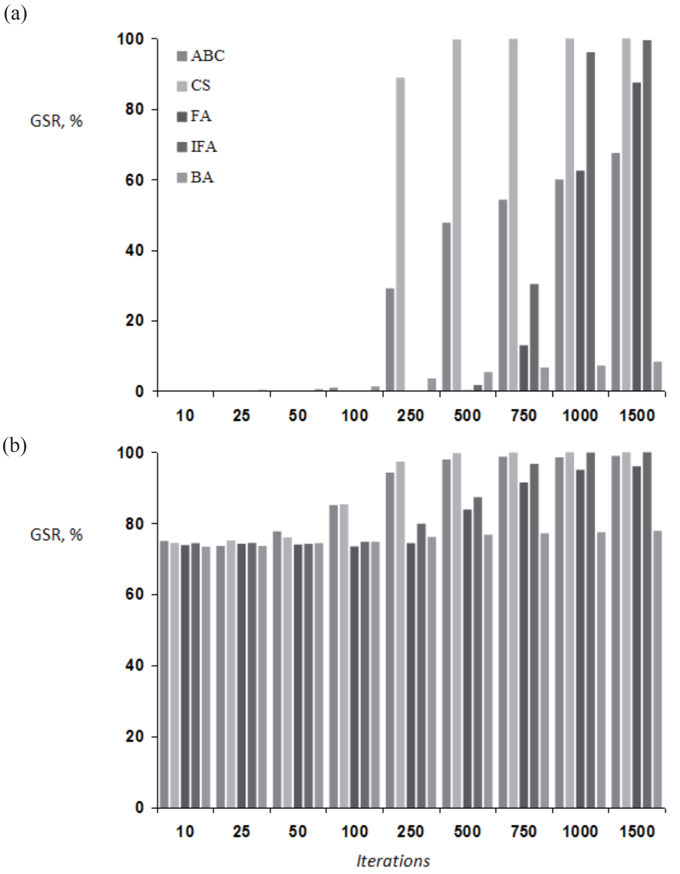


Fig. 9 Global Success Rate (GSR) versus Iterations for RPEC problems using ABC, CS, FA, IFA, and BA with SC-1: (a) bio-inspired method only and (b) bio-inspired method combined with local optimization

iterations for these problems as well. The highest GSR is 100% obtained by CS. At all iterations, CS consistently obtained best GSR, reaching almost 100% at 500 iterations. On the other hand, FA and IFA obtained better GSR at higher iterations. GSR of CS remained very low until 100 iterations when it suddenly climbed high and fast until it reached the highest GSR of 100% at 1000 iterations. In short, when comparing the bio-inspired optimization methods without the use of subsequent local optimization, CS can not be challenged with the rest of the four methods.

When the results of the stochastic global optimization after 1500 iterations followed by local optimization are compared (Fig. 9b), CS and IFA comes in top in terms of reliability with GSR of 100% compared to 99% for ABC, 96% for FA and 77.875% for BA. The performance of ABC, CS, and IFA were very good since both have reached very close to the final GSR after only 500 iterations for CS and ABC, and 750 iterations for IFA, although no significant improvement was obtained in subsequent iterations. In short, CS is the most reliable and effective to find the global optimum in relatively small number of iterations.

Results obtained on the effect of stopping criteria on the bio-inspired optimization algorithms using SC-2 with SC_{max} = 6D, 12D and 24D, for RPEC problems are summarized in Fig. 10. Note that SC_{max} values used for each RPEC problem were those used by Bonilla-Petriciolet et al. [3] so that their and the present results can be compared. We conclude that the higher the SC_{max}, the better the reliability of the algorithm is, and that the use of SC-2 gives substantially inferior GSR compared to SC-1 (Fig. 10a), especially for FA and IFA. The use of SC-2 for ABC, CS, and BA will bring about slightly inferior reliability compared to the use of SC-1 but with higher efficiency. This is not the same conclusion that can be drawn for FA and IFA as SC-1 gave the highest reliability (96 and 100%, respectively) as shown in Fig. 10a.

5.4. Comparison with the reported performance of other stochastic methods

Recently, Zhang et al. [5] reported the performance of unified bare-bones particle swarm optimization (UBBPSO), Integrated Differential Evolution (IDE) and IDE without tabu list and radius (IDE_N). They also analyzed the performance of UBBPSO, IDE and IDE_N, and compared them with other published results such as classical PSO with quasi-Newton method (PSO-CQN), classical PSO with Nelder-Mead simplex method (PSO-CNM), Simulated Annealing (SA), Genetic Algorithm (GA), and Differential Evolution with Tabu List (DETL). All these stochastic algorithms were run 100 times independently, and at the end of every run, a deterministic local optimizer was activated. Zhang et al. [5] reported that IDE gave better performance across the entire spectrum of problems. Hence, it is sufficient to compare the performance of the five bio-inspired

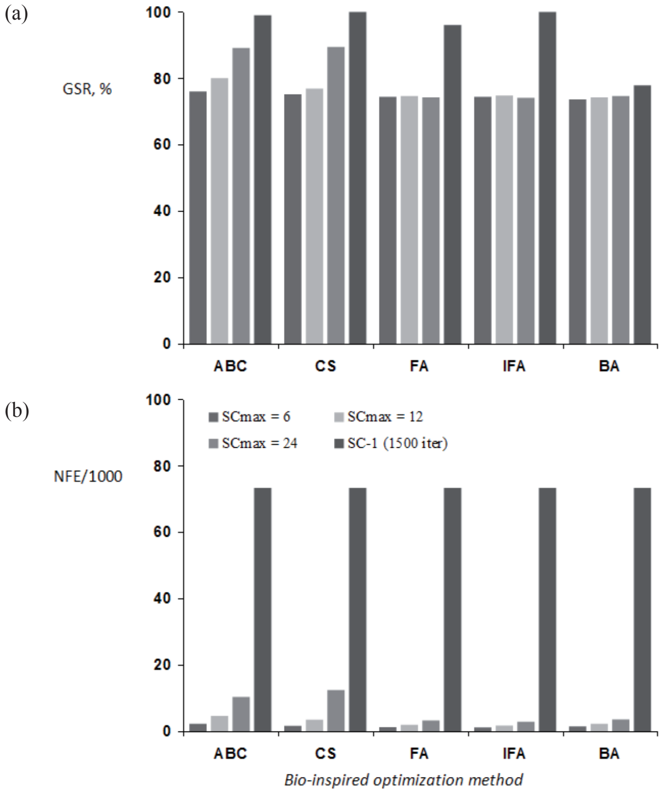


Fig. 10 (a) Global Success Rate (GSR) and (b) average NFE of ABC, CS, FA, IFA, and BA for RPEC problems using SC-2 (SC_{max} = 6D, SC_{max} = 12D and SC_{max} = 24D) and SC-1 (1500 iterations)

algorithms with IDE for the three categories of problems, with the different stopping criteria.

Figure 11 shows the average GSR of ABC, CS, FA, IFA, and BA for the 24 problems as compared with the average GSR of IDE, at different iterations. ABC shows the best convergence rate as its average GSR reaches about 80.9% after 50 iterations only. The reliability of IDE is superior to CS and slightly inferior to ABC at 50 iterations At larger iterations, the GSR of CS was the highest, reaching 99.5% at the 1500th iteration compared to 92.8% for IDE. Fig. 12 shows the average GSR of ABC, CS, FA, IFA, and BA for the 24 problems as compared with the average GSR of IDE, when SC-2 was used; IDE is superior over the other five algorithms in terms of its reliability as shown in Fig. 12a.

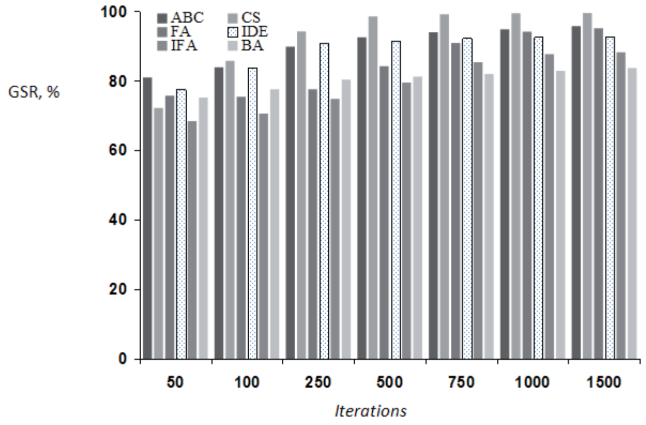


Fig. 11 Global Success Rate (GSR) of ABC, CS, FA, IFA, and BA and IDE for all problems using bio-inspired method combined with local optimization

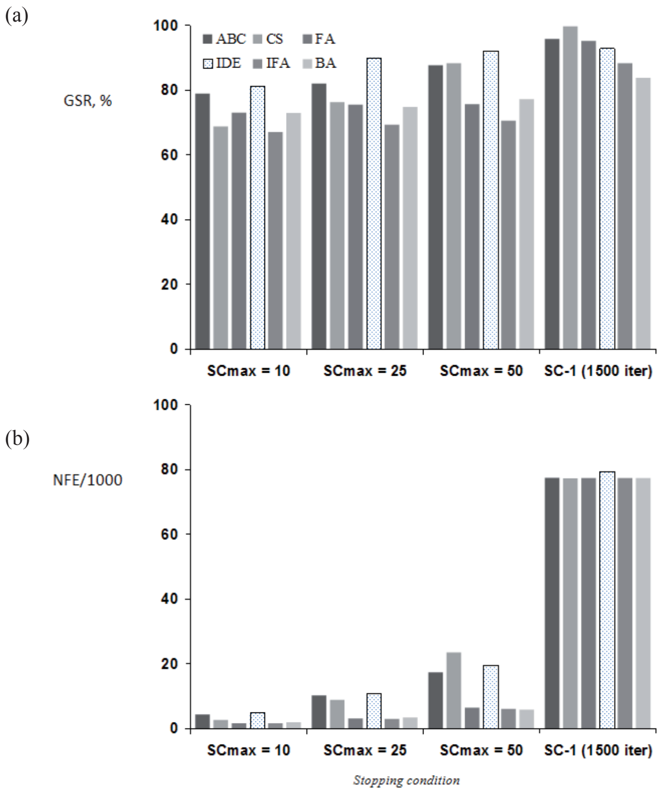


Fig. 12 (a) Global Success Rate (GSR) and (b) average NFE of ABC, CS, FA, IFA, BA and IDE for all problems using SC-2 (SC_{max} = 10, SC_{max} = 25 and SC_{max} = 50 except for RPEC problems: SC_{max} = 6, SC_{max} = 12 and SC_{max} = 24) and SC-1 (1500 iterations)

6 Conclusions

In this study, four swarm intelligence stochastic global optimization algorithms, namely, ABC, CS, IFA and BA, have been evaluated for solving the challenging phase stability, and phase and chemical equilibrium problems. Performance at different iteration levels and the effect of stopping criterion have also been analyzed. CS was found to be the most reliable technique across different problems tried at the time that it requires similar computational effort to the other methods. The stopping criterion, SC-1 gives in general 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 significantly affecting the reliability of the stochastic algorithms studied. Comparison of the performance of ABC, CS, FA, IFA and BA with the results in Zhang et al. [5] shows that IDE is generally outperformed by CS.

Table 4 Success rate (SR) and number of function evaluations of ABC, CS, FA, IFA and BA for PS problems using SCmax with NP of 10D

PS No.	SCmax	ABC		CS		FA		IFA		BA	
		SR	NFE	SR	NFE	SR	NFE	SR	NFE	SR	NFE
PS-1	10	59	683	38	634	51	517	21	389	64	510
	25	87	1,902	72	1,988	66	1,094	30	831	68	1,068
	50	99	3,504	88	4,308	79	1,988	37	1,637	79	2,094
PS-2	10	56	1,580	34	1,439	57	1,087	18	1,021	9	870
	25	20	3,148	43	4,583	32	2,083	23	1,840	11	1,688
	50	23	6,962	64	10,229	21	3,665	32	3,312	14	3,128
PS-3	10	100	1,383	100	1,485	100	712	100	748	100	727
	25	100	4,472	100	4,910	100	1,701	100	1,718	100	1,525
	50	100	8,130	100	9,660	100	3,191	100	3,416	100	2,981
PS-4	10	100	610	97	725	100	442	96	458	99	455
	25	100	1,473	97	2,100	100	1,046	96	1,033	99	999
	50	100	3,207	100	4,069	100	2,158	96	2,012	99	1,849
PS-5	10	78	2,822	8	1,625	60	1,382	8	1,232	86	1,478
	25	87	6,132	24	3,983	57	3,179	1	2,109	90	3,264
	50	93	11,321	80	15,529	38	5,902	0	3,398	90	5,829
PS-6	10	96	6,853	38	4,224	30	2,562	20	3,283	40	4,160
	25	92	14,588	79	18,811	36	5,482	18	6,034	38	7,242
	50	91	30,481	100	70,935	43	13,659	19	10,673	38	11,693
PS-7	10	100	21,263	99	8,257	99	4,536	98	4,294	99	4,396
	25	100	35,935	100	36,295	97	9,690	98	7,400	98	6,967
	50	100	41,321	100	90,854	98	18,967	98	13,749	98	11,104
PS-8	10	61	10,803	4	5,153	53	3,244	36	4,268	52	3,340
	25	84	24,371	0	7,050	60	7,127	42	8,076	49	6,377
	50	94	43,605	22	19,843	61	18,158	40	14,494	49	11,823
Total NFE – Average GSR		84	286,548	66	328,689	68	113,569	51	97,425	70	95,567

Table 5 Success rate (SR) and number of function evaluations (NFE) of CMA-ES, SCE and FA for PEC problems using SCmax with NP of 10D

PEC No.	SC_{max}	ABC		CS		FA		IFA		BA	
		SR	NFE	SR	NFE	SR	NFE	SR	NFE	SR	NFE
PEC-1	10	72	1,234	91	930	51	499	57	499	45	565
	25	86	3,386	100	2,858	82	1,127	84	1,127	68	1,439
	50	98	4,879	100	7,793	93	2,068	97	2,068	84	2,784
PEC-2	10	14	1,070	43	1,390	4	829	2	971	17	1,078
	25	14	2,675	98	6,559	0	1,832	1	1,866	29	2,023
	50	14	6,347	100	15,983	0	3,449	2	3,690	37	3,553
PEC-3	10	100	1,392	100	1,297	100	677	100	703	100	832
	25	100	5,908	100	5,539	100	1,515	100	1,569	100	1,659
	50	100	20,599	100	14,529	100	3,339	100	3,411	100	3,021
PEC-4	10	59	533	65	618	55	463	67	494	52	546
	25	75	1,725	92	3,224	86	1,187	78	1,129	62	1,181
	50	90	5,487	99	7,292	97	2,533	86	2,279	74	2,221
PEC-5	10	91	3,561	34	2,355	95	1,890	89	1,849	96	2,212
	25	91	6,950	23	5,088	97	3,337	92	3,566	93	3,680
	50	95	11,878	50	14,407	89	5,887	93	6,876	97	5,911
PEC-6	10	97	7,340	99	5,809	100	2,856	100	2,869	100	4,316
	25	91	16,928	92	19,822	100	5,462	100	5,746	95	8,767
	50	92	29,166	99	55,674	100	10,323	99	15,065	94	17,378
PEC-7	10	100	14,073	100	6,148	100	3,078	100	3,596	100	5,491
	25	100	53,746	100	27,250	100	7,048	100	6,554	100	8,977
	50	100	66,766	100	107,251	100	17,608	100	11,471	100	13,524
PEC-8	10	100	11,257	97	7,350	100	3,455	100	3,326	100	3,333
	25	100	24,227	93	32,890	100	6,409	100	6,964	100	6,435
	50	100	41,234	100	15,100	100	15,100	100	23,536	100	11,786
Total NFE – Average GSR		82	342,358	86	367,155	81	101,974	81	111,223	81	112,713

Nomenclature

D	Dimension of the problem
F, F_{obj}	Objective function
$\mathbf{g}_*$	Global minimum
I	Brightness of a firefly in FA
i, j	Index of the component, or index of the firefly in FA
K	Index of the problem dimension in FA
Np	Number of problems
$\mathbf{n}_{ref}$	Column vector of moles of each of the reference components
ν	Position of new food source in ABC
X	The decision variable vector

Greek letters

β	Transformed decision variables used instead of mole fractions, or attractiveness of a firefly in FA
ε	Vector of random numbers in FA
ϕ	Fraction of top fireflies to be utilized in the move
φ	Random number in the range [0,1] in ABC
Φ	Random number in the range [-1,1] in ABC

Subscripts

F	Feed
I	Index for the components in the mixture

K	Index for the iterations in FA
min	Minimum value
O	Initial value of the parameter
y	At composition y
z	At composition z

Superscripts

0	Pure component

References

[1] Gupta, S., Bhardwaj, S., Bhatia, P. K. "A Reminiscent Study of Nature Inspired Computation." *International Journal of Advanced Engineering Technology.* 1 (2). pp. 117–125. 2011.

[2] Bonilla-Petriciolet, A., Rangaiah, G. P., Segovia-Hernández, J. G. "Constrained and unconstrained Gibbs free energy minimization in reactive systems using genetic algorithm and differential evolution with tabu list." *Fluid Phase Equilibra.* 300 (1–2). pp. 120–134. 2011. DOI: 10.1016/j.fluid.2010.10.024

[3] Bhargava, V., Fateen, S. E. K., Bonilla-Petriciolet, A. "Cuckoo Search: A new nature-inspired optimization method for phase equilibrium calculations." *Fluid Phase Equilibra.* 337. pp. 191–200. 2013.

[4] Fateen, S.-E. K, Bonilla-Petriciolet, A. "Intelligent Firefly Algorithm for Global Optimization." In: Yang, X. (ed).: *Cuckoo Search and Firefly Algorithm*. Berlin / Heidelberg: Springer. 2013.

[5] Fateen, S.-E. K., Bonilla-Petriciolet, A., Rangaiah, G. P. "Evaluation of Covariance Matrix Adaptation Evolution Strategy, Shuffled Complex Evolution and Firefly Algorithms for phase stability, phase equilibrium and chemical equilibrium problems." *Chemical Engineering Reserach and Design.* 90 (12). pp. 2051–2071. 2012. DOI: 10.1016/j.cherd.2012.04.011

[6] Fateen S.-E. K., Bonilla-Petriciolet, A. "A note on effective phase stability calculations using a Gradient-Based Cuckoo Search algorithm." *Fluid Phase Equilibra.* 375. pp. 360–366. 2014. DOI: 10.1016/j.fluid.2014.05.009

[7] Elnabawy, A. O., Fateen, S.-E. K., Bonilla-Petriciolet, A. "Phase Stability Analysis and Phase Equilibrium Calculations in Reactive and Nonreactive Systems Using Charged System Search Algorithms." *Industrial & Engineering Chemistry Research.* 53 (6). pp. 2382–2395. 2014. DOI: 10.1021/ie403123b

[8] Fateen S.-E. K., Bonilla-Petriciolet, A. "Unconstrained Gibbs Free Energy Minimization for Phase Equilibrium Calculations in Nonreactive Systems, Using an Improved Cuckoo Search Algorithm." *Industrial & Engineering Chemistry Research.* 53 (26). pp. 10826–10834. 2014. DOI: 10.1021/ie403123b

[9] Srinivas, M., Rangaiah, G. "A study of differential evolution and tabu search for benchmark, phase equilibrium and phase stability problems." *Computers & Chemical Engineering.* 31 (7). pp. 760–772. 2007. DOI: 10.1016/j.compchemeng.2006.07.015

[10] Zhang, H., Bonilla-Petriciolet, A., Rangaiah, G. "A review on global optimization methods for phase equilibrium modeling and calculations." *The Open Thermodynamics Journal.* 5(1). pp. 71–92. 2011. DOI: 10.2174/1874396x01105010071

[11] Yang, X. S. "Firefly Algorithm." In: *Nature-Inspired Metaheuristic Algorithms*. pp. 79–90. 2007.

[12] Gandomi, A. H., Yang, X.-S., Alavi, A. H. "Mixed variable structural optimization using Firefly Algorithm." *Computers & Structures.* 89 (23–24). pp. 2325–2336. 2011. DOI: 10.1016/j.compstruc.2011.08.002

[13] Sayadi, M. K., Ramezanian, R., Ghaffari-Nasab, N. "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems." *International Journal of Industrial Engineering Computations.* 1 (1). pp. 1–10. 2010. DOI: 10.5267/j.ijiec.2010.01.001

[14] Giannakouris, G., Vassiliadis, V., Dounias, G. "*Experimental Study on a Hybrid Nature-Inspired Algorithm for Financial Portfolio Optimization Artificial Intelligence: Theories, Models and Applications*." 6040. Berlin / Heidelberg, 2010. pp. 101–111.

[15] Karaboga, D., Basturk, B. "On the performance of artificial bee colony (ABC) algorithm." *Applied Soft Computing.* 8 (1). pp. 687–697. 2008. DOI: 10.1016/j.asoc.2007.05.007

[16] Yang, X. S., Deb, S. "Cuckoo search via Lévy flights." *2009 World Congress on Nature & Biologically Inspired Computing.* pp. 210–214. 2009. DOI: 10.1109/nabic.2009.5393690

[17] Altringham, J. D. "*Bats: Biology and Behaviour.*" Oxford Univesity Press. 1996.

[18] Yang, X. S. "A new metaheuristic bat-inspired algorithm." *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74. 2010.

[19] Rangaiah, G. P. "Evaluation of genetic algorithms and simulated annealing for phase equilibrium and stability problems." *Fluid Phase Equilibra.* 187–188. pp. 83–109. 2001. DOI: 10.1016/s0378-3812(01)00528-3

[20] Burgos-Solórzano, G. I., Brennecke, J. F., Stadtherr, M. A. "Validated computing approach for high-pressure chemical and multiphase equilibrium." *Fluid Phase Equilibra.* 219 (2). pp. 245–255. 2004. DOI: 10.1016/j.fluid.2003.12.013

[21] Seider, W. D., Widagdo, S. "Multiphase equilibria of reactive systems." *Fluid Phase Equilibra.* 123 (1–2). pp. 283–303. 1996. DOI: 10.1016/s0378-3812(96)90039-4