

# Hybrid Particle Swarm-Ant Colony Algorithm to Describe the Phase Equilibrium of Systems Containing Supercritical Fluids with Ionic Liquids

Juan A. Lazzús\*

*Departamento de Física, Universidad de La Serena, Casilla 554, La Serena, Chile.*

Received 20 March 2012; Accepted (in revised version) 19 July 2012

Communicated by Bo Li

Available online 18 October 2012

---

**Abstract.** Based on biologically inspired algorithms, a thermodynamic model to describe the vapor-liquid equilibrium of binary complex mixtures containing supercritical fluids and ionic liquids, is presented. The Peng-Robinson equation of state with the Wong-Sandler mixing rules are used to evaluate the fugacity coefficient on the systems. Then, a hybrid particle swarm-ant colony optimization was used to minimize the difference between calculated and experimental bubble pressure, and calculate the binary interaction parameters for the excess Gibbs free energy of all systems used. Simulations are carried out in nine systems with imidazolium-based ionic liquids. The results show that the bubble pressures were correlated with low deviations between experimental and calculated values. These deviations show that the proposed hybrid algorithm is the preferable method to describe the phase equilibrium of these complex mixtures, and can be used for other similar systems.

**AMS subject classifications:** 80M50, 92-08, 82D15, 82B26

**PACS:** 51.30.+i, 64.75.Cd, 02.60.Pn

**Key words:** Particle swarm optimization, ant colony optimization, vapor-liquid equilibrium, ionic liquids, supercritical fluids, Peng-Robinson equation of state.

---

## 1 Introduction

Phase equilibrium data of mixtures containing ionic liquids are necessary for further development of some separation processes [1]. Blanchard *et al.* [2] described several potential applications of supercritical fluids with ionic liquids. The gas solubility data provides important information for the characterization of solute-solvent interactions and so contribute to understand the mechanisms of dissolution. From a practical point of view, gas solubility can be useful in the calculation of vapor-liquid equilibrium [3].

---

\*Corresponding author. *Email address:* jlazzus@dfuls.cl (J. A. Lazzús)

One of the common approaches used in the literature to correlate and predict phase equilibrium requires an equation of state that well relates the variables temperature, pressure and volume and appropriate mixing rules to express the dependence of the equation of state parameters on the concentration [1]. The existing methods to solve phase equilibrium systems obtain only local solutions. It has been demonstrated that for cases of systems containing supercritical fluids, the optimum values of the interaction parameters depend on the searching interval and on the initial value of used interaction parameters [4]. Then, the parameter optimization procedures are very important in engineering, industrial, and chemical process for development of mathematical models, since design, optimization and advanced control of bioprocesses depend on model parameter values obtained from experimental data [1].

The aim of optimization is to determine the best-suited solution to a problem under a given set of constraints. Mathematically, an optimization problem involves a fitness function describing the problem, under a set of constraints representing the solution space for the problem. The optimization problem, now-a-days, is represented as an intelligent search problem, where one or more agents are employed to determine the optimum on a search landscape [5]. Modern optimization techniques have aroused great interest among the scientific and technical community in a wide variety of fields recently, because of their ability to solve problems with non-linear and non-convex dependence of design parameters [6].

Thus, the use of heuristic optimization methods, such particle swarm optimization [7] and ant colony optimization [8], for the parameter estimation is very promising [1]. These biologically-derived methods represent an excellent alternative to find a global optimum for phase equilibrium calculations.

In this work, nine binary vapor-liquid phase systems containing supercritical fluids and ionic liquids were evaluated using a hybrid algorithm based on particle swarm optimization and ant colony optimization. The complete program was used to calculate the binary interaction parameters of these complex mixtures by minimization of the difference between calculated and experimental data.

## 2 Thermodynamic model

As known, the phase equilibrium problem to be solved consists of the calculation of some variables of the set  $T$ - $P$ - $x$ - $y$  (temperature, pressure, liquid-phase concentration and vapor-phase concentration, respectively), when some of them are known. For a vapor-liquid mixture in thermodynamic equilibrium, the temperature and the pressure are the same in both phases, and the remaining variables are defined by the material balance and the "*fundamental equation of phase equilibrium*". The application of this fundamental equation requires the use of thermodynamic models which normally include binary interaction parameters [9].

The classical thermodynamic models commonly used in the literature to treat these

mixtures at low pressure required a great amount of binary parameters to be determined from experimental data [10]. These binary parameters must be determined using experimental data for binary systems. Theoretically, once these binary parameters are known one could predict the behavior of multicomponent mixtures using standard thermodynamic relations and thermodynamics models [9].

The fundamental equation of vapor-liquid equilibrium can be expressed as the equality of fugacities of each component in the mixture in both phases [10]:

$$\bar{f}_i^L = \bar{f}_i^V, \tag{2.1}$$

where the superscripts  $L$  and  $V$  represent liquid and vapor, respectively.

The fugacity of a component in the vapor phase is usually expressed through the fugacity coefficient  $\bar{\phi}_i^V$ :

$$\bar{f}_i^V = y_i \bar{\phi}_i^V P. \tag{2.2}$$

And the fugacity of a component in the liquid phase is expressed through either the fugacity coefficient  $\bar{\phi}_i^L$  or the activity coefficient  $\gamma_i$ :

$$\bar{f}_i^L = x_i \bar{\phi}_i^L P, \tag{2.3}$$

$$\bar{f}_i^L = x_i \gamma_i f_i^0. \tag{2.4}$$

In these equations,  $y_i$  is the mole fraction of component in the vapor phase,  $x_i$  is the mole fraction of component in the liquid phase, and  $P$  is the pressure. The fugacity is related to the temperature, the pressure, the volume and the concentration through a standard thermodynamic relation [5].

If the fugacity coefficient is used in both phases, the method of solution of the phase equilibrium problem is known as "*the equation of state method*". Then, equation of state (EoS) and a set of mixing rules are needed, to express the fugacity coefficient as function of the temperature, the pressure and the concentration [9]. Modern EoS methods include an excess Gibbs free energy model ( $G^E$ ) in the mixing rules of the EoS, giving origin to the so-called "*equation of state +  $G^E$  model*" [10]. This means that an activity coefficient model ( $\gamma$ ) is used to describe the complex liquid phase, and the fugacity coefficient ( $\phi$ ) is calculated using a simple equation of state. If the fugacity coefficient is used for the vapor phase and the activity coefficient is used for the liquid phase the equilibrium problem is known as "*the gamma-phi method*" ( $\gamma$ - $\phi$ ) [9]. This method has given acceptable results for some systems [1,5].

From the relation between the fugacity, the Gibbs free energy, and an EoS, the fugacity in a vapor can be calculated as:

$$\ln \left[ \frac{\bar{f}_i^V(T, P, y_i)}{y_i P} \right] = \ln \bar{\phi}_i, \tag{2.5}$$

$$\ln \bar{\phi}_i = \frac{1}{RT} \int_{V=\infty}^V \left[ \frac{RT}{V} - \left( \frac{\partial P}{\partial N_i} \right)_{T, V, N_{i \neq j}} \right] \partial V - \ln Z^V, \tag{2.6}$$

where  $V$  is the total volume, and  $Z=PV/(RT)^{-1}$  is the compressibility factor calculated from an EoS, and  $\underline{V}$  is the molar volume of the mixture [10].

The most common EoS used for the correlation of phase equilibria in mixtures at high and low pressure are the cubic equations derived from van der Waals EoS [11]; among these, the Peng-Robinson equation has proven to combine the simplicity and accuracy required for the prediction and correlation of volumetric and thermodynamic properties of fluids [12].

The Peng-Robinson EoS was expressed as follows [12]:

$$P = \frac{RT}{\underline{V}-b} + \frac{a}{\underline{V}(\underline{V}+b)+b(\underline{V}-b)}, \quad (2.7)$$

with

$$a = 0.457235 \frac{R^2 T_c^2}{P_c} \alpha(T_r), \quad (2.8)$$

$$b = 0.077796 \frac{RT_c}{P_c}, \quad (2.9)$$

$$\alpha(T_r) = [1 + \kappa(1 - \sqrt{T_r})]^2, \quad (2.10)$$

$$\kappa = 0.37646 + 1.54226\omega - 0.26992\omega^2, \quad (2.11)$$

where  $T_r = T/T_c$  is the reduced temperature. In this form, the Peng-Robinson EoS is completely predictive once the constants (critical temperature  $T_c$ , critical pressure  $P_c$ , and acentric factor  $\omega$ ) are given. Consequently, this equation is a two-parameter EoS ( $a$  and  $b$ ) that depends upon the three constants ( $T_c$ ,  $P_c$ , and  $\omega$ ) [10].

For mixtures, the parameters  $a$  and  $b$  are expressed as functions of the concentration of the different components in the mixture, through the so-called mixing rules [9]. Until recent years, most of the applications of EoS to mixtures considered the use of the classical van der Waals mixing rules, with the inclusion of an interaction parameter for the force constant  $a$  and volume constant  $b$ . The Peng-Robinson EoS for a mixture is:

$$P = \frac{RT}{\underline{V}-b_m} + \frac{a_m}{\underline{V}(\underline{V}+b_m)+b_m(\underline{V}-b_m)}. \quad (2.12)$$

The classical van der Waals mixing rules are:

$$a_m = \sum_i \sum_j x_i x_j a_{ij}, \quad (2.13)$$

$$b_m = \sum_i \sum_j x_i x_j b_{ij}, \quad (2.14)$$

and the combining rules for  $a_{ij}$  and  $b_{ij}$ , with interaction parameters for the force and volume constants, are:

$$a_{ij} = \sqrt{a_i a_j} (1 - k_{ij}), \quad (2.15)$$

$$b_{ij} = \frac{b_i + b_j}{2} (1 - l_{ij}). \quad (2.16)$$

The parameters  $k_{ij}$  and  $l_{ij}$  in the above combining rules for the equation of state are usually calculated by regression analysis of experimental phase equilibrium data.

But, the modern equation of state includes an excess Gibbs free energy model in the mixing rules of the EoS. Thus, the connection between equations of state + excess Gibbs free energy, seem to be the most appropriate for modeling complex mixtures [1].

The Wong-Sandler mixing rule is an example of these types of mixing rules, and can be summarized as follows [13]:

$$b_m = \frac{\sum_i^N \sum_j^N x_i x_j (b - \frac{a}{RT})_{ij}}{1 - \sum_i^N \frac{x_i a_i}{b_i RT} + \frac{A_\infty^E(x)}{\Omega RT}}, \quad (2.17)$$

$$\left(b - \frac{a}{RT}\right)_{ij} = \frac{1}{2}(b_i + b_j) - \frac{\sqrt{a_i a_j}}{RT}(1 - k_{ij}), \quad (2.18)$$

$$a_m = b_m \left( \sum_i^N \frac{x_i a_i}{b_i} + \frac{A_\infty^E(x)}{\Omega} \right). \quad (2.19)$$

In these equations  $a_m$  and  $b_m$  are the equation of state constants with  $k_{ij}$  as adjustable parameter,  $\Omega = 0.34657$  for the PR EoS, and  $A_\infty^E(x)$  is calculated assuming that  $A_\infty^E(x) \approx A_0^E(x) \approx G_0^E(x)$ .

For a binary mixture:

$$b_m = \frac{x_1^2 (b - \frac{a}{RT})_1 + 2x_1 x_2 (b - \frac{a}{RT})_{12} + x_2^2 (b - \frac{a}{RT})_2}{1 - \frac{x_1 a_1}{b_1 RT} - \frac{x_2 a_2}{b_2 RT} + \frac{G_0^E(x)}{\Omega RT}}, \quad (2.20)$$

$$\left(b - \frac{a}{RT}\right)_{12} = \frac{1}{2}(b_1 + b_2) - \frac{\sqrt{a_1 a_2}}{RT}(1 - k_{12}), \quad (2.21)$$

$$a_m = b_m \left( \frac{x_1 a_1}{b_1 RT} + \frac{x_2 a_2}{b_2 RT} + \frac{A_0^E(x)}{\Omega} \right). \quad (2.22)$$

The fugacity coefficient expression for species  $i$  in a mixture for the Wong-Sandler equations is [10]:

$$\begin{aligned} \ln \bar{\phi}_i = & \left( \frac{\partial N b}{\partial N_i} \right)_{T, N_{i \neq j}} \frac{(Z-1) - \ln(Z-B)}{b} \\ & + \frac{1}{2\sqrt{2}bRT} \left[ \frac{1}{N} \left( \frac{\partial N^2 a}{\partial N_i} \right)_{T, N_{i \neq j}} - \frac{a}{b} \left( \frac{\partial N b}{\partial N_i} \right)_{T, N_{i \neq j}} \right] \ln \left[ \frac{Z + (1 - \sqrt{2}B)}{Z + (1 + \sqrt{2}B)} \right]. \end{aligned} \quad (2.23)$$

The partial derivative terms are:

$$\left( \frac{\partial N b}{\partial N_i} \right)_{T, N_{i \neq j}} = \frac{1}{1-D} \left( \frac{1}{N} \frac{\partial N^2 Q}{\partial N_i} \right) - \frac{Q}{(1-D)^2} \left[ 1 - \left( \frac{\partial N D}{\partial N_i} \right) \right], \quad (2.24)$$

$$\frac{1}{N} \left( \frac{\partial N^2 a}{\partial N_i} \right) = RT D \left( \frac{\partial N b}{\partial N_i} \right) + RT b \left( \frac{\partial N D}{\partial N_i} \right), \quad (2.25)$$

with

$$Q = \sum_i \sum_j x_i x_j \left( b - \frac{a}{RT} \right)_{ij}, \quad (2.26)$$

$$D = \frac{G_0^E(x_i)}{\Omega RT} + \sum_i x_i \frac{a_i}{b_i RT}, \quad (2.27)$$

and

$$\left( \frac{1}{N} \frac{\partial N^2 Q}{\partial N_i} \right) = 2 \sum_j x_j \left( b - \frac{a}{RT} \right)_{ij}, \quad (2.28)$$

$$\frac{\partial ND}{\partial N_i} = \frac{a_i}{b_i RT} + \frac{\ln \gamma_i}{\Omega}, \quad (2.29)$$

$$\ln \gamma_i = \frac{1}{RT} \left( \frac{\partial N G_0^E(x_i)}{\partial N_i} \right)_{T, N_{i \neq j}}. \quad (2.30)$$

The excess Gibbs free energy  $G_0^E(x)$  in the mixing rules is calculated using an appropriate liquid-phase model. In this work,  $G_0^E(x)$  has been calculated using the van Laar model that has been shown to perform well in high pressure phase equilibrium calculations [1].

The van Laar model for  $G_0^E(x)$  is described by the following equation [9]:

$$\frac{G_0^E}{RT} = \sum_i x_i \frac{\sum_j x_j A_{ij}}{1 - x_j} \left[ 1 - \frac{x_i \sum_j x_j A_{ij}}{x - i \sum_j x_j A_{ij} + (1 - x_i) x_i \sum_j x_j A_{ij}} \right]^2. \quad (2.31)$$

For a binary mixture, the model reduces to:

$$G_0^E = \frac{A_{12} RT x_1 x_2}{x_1 \left( \frac{A_{12}}{A_{21}} \right) + x_2}. \quad (2.32)$$

Thus, the problem is reduced here to determine the interaction parameters  $A_{12}$ ,  $A_{21}$ , and the  $k_{12}$  parameter included in the combining thermodynamic model (PR-WS-VL), using available high pressure  $T$ - $P$ - $x$  data of vapor-liquid phase equilibrium of complex mixtures.

These optimal interaction parameters were determined by minimizing the following objective function in data regression, using a hybrid algorithm based on particle swarm optimization and ant colony optimization:

$$\min F = \frac{100}{N_D} \sum_{i=1}^{N_D} \left| \frac{P^{calc} - P^{exp}}{P^{exp}} \right|_i, \quad (2.33)$$

where  $N_D$  is the number of points in the experimental data set and  $P$  is the pressure of the ionic liquid in the vapor phase, the superscript denotes the experimental (*exp*) data point and calculated (*calc*) values.

### 3 Hybrid optimization technique

The hybrid algorithm was developed with particle swarm optimization and ant colony optimization. Particle swarm optimization is one of the recent meta-heuristic techniques proposed by Kennedy and Eberhart [7]. Ant colony optimization is an algorithm based on the foraging behavior of ants, and has been first introduced by Dorigo [8].

#### 3.1 Particle swarm optimization

Particle swarm optimization is a stochastic technique motivated by the behavior of a flock of birds or the sociological behavior of a group of people [7].

The particle swarm algorithm is initialized with a population of random particles and the algorithm searches for optima by updating generations [14]. In a particle swarm system, each particle is "flown" through the multidimensional search space, adjusting its position in search space according to its own experience and that of neighboring particles. The particle therefore makes use of the best position encountered by itself and that of its neighbors to position itself toward an optimal solution [4–6]. The performance of each particle is evaluated using a predefined fitness function, which encapsulates the characteristics of the optimization problem [15].

Let  $s$  and  $v$  denote a particle position and its corresponding velocity in a search space, respectively [7]. Therefore, the  $\lambda$ -th particle is represented in the  $n$ -dimensional search space as:

$$s^\lambda = (s_1^\lambda, s_2^\lambda, \dots, s_n^\lambda). \quad (3.1)$$

And the current velocity of the  $\lambda$ -th particle is represented as:

$$v^\lambda = (v_1^\lambda, v_2^\lambda, \dots, v_n^\lambda). \quad (3.2)$$

Let the current personal best position of the particle and  $F(s)$  be the target function which will be minimized.

$$p^\lambda = (p_1^\lambda, p_2^\lambda, \dots, p_n^\lambda). \quad (3.3)$$

Then the best position  $p^\lambda$  is determined by:

$$p_{t+1}^\lambda = \begin{cases} p_t^\lambda, & \text{if } F(s_{t+1}^\lambda) \geq F(p_t^\lambda), \\ s_{t+1}^\lambda, & \text{if } F(s_{t+1}^\lambda) \leq F(p_t^\lambda). \end{cases} \quad (3.4)$$

Let  $t$  be a time instant. The new particle position is computed by adding the velocity vector to the current position:

$$s_{t+1}^\lambda = s_t^\lambda + v_{t+1}^\lambda, \quad (3.5)$$

where  $s_{t+1}^\lambda$  is the particle position at time instant  $t$ , and  $v_{t+1}^\lambda$  is the new velocity at time  $t+1$ .

The velocity update equation is given by:

$$v_{t+1}^\lambda = w_t v_t^\lambda + c_1 r_1 (p_t^\lambda - s_t^\lambda) + c_2 r_2 (p_t^g - s_t^\lambda), \quad (3.6)$$

where  $w$  is the inertia weight,  $c_1$  and  $c_2$  are the acceleration constants, and  $r_1, r_2$  are element from two random sequences in the range  $(0,1)$ . The current position of the particle is determined by  $s_t^\lambda$ ;  $p_t^\lambda$  is the best one of the solutions that this particle has reached, is the best one of the all solutions that the particles have reached [4–6].

The variable  $w$  is responsible for dynamically adjusting the velocity of the particles, so it is responsible for balancing between local and global search, hence requiring fewer iterations for the algorithm to converge [16]. A low value of inertia weight implies a local search, while a high value leads to a global search. Applying a large inertia weight at the start of the algorithm and making it decay to a small value through the particle swarm optimization execution makes the algorithm search globally at the beginning of the search, and search locally at the end of the execution [1]. The following weighting function is used in Eq. (3.6):

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}} t, \quad (3.7)$$

where the subscript *min* and *max* are the minimum and maximum values selected for these parameters. Generally, the value of each component in  $v$  can be clamped to the range  $[-v_{\max}, v_{\max}]$  to control the excessive roaming of particles outside the search space [14, 15]. After calculating the velocity, the particle swarm algorithm performs repeated applications of the update equations above until a specified number of iteration has been exceeded, or until the velocity updates are close to zero [4–6]. The scheme of the particle swarm algorithm is presented in detail in Table 1.

Table 1: Scheme of the particle swarm algorithm.

| Step | Description  |
|------|--|
| 01   | Initialize algorithm. Set constants: $t_{\max}, v_{\max}, w, c_1, c_2$   |
| 02   | Randomly initialize the swarm positions $s_0^\lambda \in \mathbb{R}^n$ for $\lambda = 1, \dots, \rho$  |
| 03   | Randomly initialize particle velocities $s_0^\lambda$ for $\lambda = 1, \dots, \rho$   |
| 04   | Set $t = 1$  |
| 05   | Evaluate function value $F_t^\lambda$ using design space coordinates $s_t^\lambda$ :<br>If $F_t^\lambda \leq F_{best}^\lambda$ then $F_{best}^\lambda = F_t^\lambda, F_t^\lambda = F_t^\lambda$<br>If $F_t^g \leq F_{best}^g$ then $F_{best}^g = F_t^g, F_t^g = F_t^g$ |
| 06   | If stopping condition is satisfied then stop algorithm   |
| 07   | Update all particle positions $v_t^\lambda$ for $\lambda = 1, \dots, \rho$   |
| 08   | Update all particle positions $s_t^\lambda$ for $\lambda = 1, \dots, \rho$   |
| 09   | Otherwise set $t = t + 1$ goes to step 05  |

### 3.2 Ant colony optimization

The basic idea of ant colony optimization is to imitate the cooperative behavior of ant colonies [8]. As soon as an ant finds a food source, it evaluates it and carries some food back to the nest [17].

Ants are insects which live together. Since they are blind animals, they find the shortest path from nest to food with the aid of pheromone. The pheromone is the chemical material deposited by ants, which serves as critical communication media among ants, thereby guiding the determination of the next movement. On the other hand, ants find the shortest path based on intensity of pheromone deposited on different paths [18]. Generally, intensity of pheromone and the length of the path are used to simulate ant system. In ant colony algorithm, the probability with which an ant  $\lambda$  chooses to go from city  $u$  to city  $u'$  is:

$$p_t^\lambda = \begin{cases} \frac{[\tau_{uu'}(t)]^{\gamma_2} [1/L_{uu'}]^{\gamma_1}}{\sum_{l \in N_u^\lambda} [\tau_{ul}(t)]^{\gamma_2} [1/L_{ul}]^{\gamma_1}}, & \text{if } u' \in N_u^\lambda, \\ 0, & \text{if otherwise,} \end{cases} \quad (3.8)$$

where  $\tau_{uu'}$  and  $L_{uu'}$  are the intensity of pheromone and the length of the path between cities  $u$  and  $u'$  respectively.  $\gamma_1$  and  $\gamma_2$  are the control parameters for determining the weight of the trail intensity and the length of the path, respectively.  $N_u^\lambda$  is the set of neighbors of city  $u$  for the  $\lambda$ -th ant. After selecting the next path, the trail intensity of pheromone is updated as:

$$\tau_{uu'}(t+1) = (1-\mu)\tau_{uu'}(t) + \Delta\tau_{uu'}(t), \quad (3.9)$$

with

$$\tau_{uu'}(t) = \begin{cases} \frac{1}{Lm}, & \text{if } (uu') \in \text{global-best-tour,} \\ 0, & \text{if otherwise.} \end{cases} \quad (3.10)$$

In the above equation  $0 \leq \mu \leq 1$ , is the pheromone trial evaporation rate [19].  $\Delta\tau_{uu'}$  is the amount of pheromone trail added to  $\tau_{uu'}$  by ants [17]. And  $Lm$  is the length of the global best tour [18]. The scheme of the particle swarm algorithm is presented in detail in Table 2.

### 3.3 Particle swarm-ant colony optimization (PSO-ACO)

The hybrid algorithm is based on the common characteristics of particle swarm optimization and ant colony optimization, like, survival as a swarm (colony) by coexistence and cooperation, individual contribution to food searching by a particle (ant) by sharing information locally and globally in the swarm (colony) between particles (ants), etc. The implementation of PSO-ACO algorithm consists of two stages. In the first stage, it applies particle swarm optimization, while ant colony optimization is implemented in the second stage. Ant colony works as a local search, wherein, ants apply pheromone-guided

Table 2: Scheme of the ant colony algorithm.

| Step | Description  |
|------|--|
| 01   | Initialize algorithm:<br>Assign to each $\lambda$ ant an initial solution            |
| 02   | Generate a pheromone trail   |
| 03   | Set $t = 1$  |
| 04   | Evaluate function value $F_t^\lambda$ :<br>Create the solution using pheromone trail |
| 05   | If stopping condition is satisfied then stop algorithm                               |
| 06   | Update pheromone trails according to solutions created                               |
| 07   | Otherwise set $t = t + 1$ goes to step 04  |

mechanism to refine the positions found by particles in the particle swarm stage [20]. Thus, the particles update their positions and velocities as follows:

$$s_{t+1}^\lambda = s_t^\lambda + v_{t+1}^\lambda, \quad (3.11)$$

$$v_{t+1}^\lambda = w_t v_t^\lambda + c_1 r_1 (p_t^\lambda - s_t^\lambda) + c_2 r_2 (p_t^g - s_t^\lambda) + c_3 r_3 (R_t^\lambda - s_t^\lambda), \quad (3.12)$$

where  $w$  is the inertia weight,  $c_1$  and  $c_2$  are the acceleration constants,  $c_3$  is the passive congregation coefficient, and  $r_1, r_2, r_3$  are element from three random sequences in the range  $(0,1)$ . The current position of the particle is determined by  $s_t^\lambda$ , and  $v_{t+1}^\lambda$  is the new velocity at time  $t+1$ ;  $p_t^\lambda$  is the best one of the solutions that this particle has reached,  $p_t^g$  is the best one of the all solutions that the particles have reached [21].

In PSO-ACO, a simple pheromone-guided mechanism of ant colony optimization is proposed to apply as local search [22]. The ant colony algorithm handles  $\rho$  ants equal to the number of particles in the swarm. Each ant  $\lambda$  generates a solution  $z_t^\lambda$  around  $p_t^g$  the global best-found position among all particles in the swarm up to iteration count  $t$  as:

$$z_t^\lambda = \mathcal{N}(p_t^g, \sigma), \quad (3.13)$$

where  $\mathcal{N}(p_t^g, \sigma)$  denotes a random number obtained by Gaussian function with mean value  $p_t^g$  and variance  $\sigma$  defined as [21]:

$$\sigma = (Y_{\max} - Y_{\min}) \cdot \eta, \quad (3.14)$$

where  $\eta$  is used to control the step size. In the standard ant colony algorithms, the probability of selecting a path with more pheromone is greater than other paths [22]. Similarly, in the Gaussian functions, the probability of selecting a solution in the neighborhood of  $p_t^g$  is greater than the others [21].

In this hybrid algorithm, the objective function value  $F(z_t^\lambda)$  is computed and the current position of ant  $\lambda$ ,  $z_t^\lambda$  is replaced with the position  $s_t^\lambda$ , the current position of particle  $\lambda$  in the swarm, if  $F(s_t^\lambda) \geq F(z_t^\lambda)$  and current ant is in the feasible space [20]. This simple pheromone-guided mechanism considers, there is highest density of trails (single

Table 3: Scheme of the hybrid PSO-ACO algorithm.

| Step | Description  |
|------|--|
| 01   | Set $t = 0$  |
| 02   | Randomly initialize positions and velocities of all particles  |
| 03   | FOR (each particle $\lambda$ in the initial population)  |
| 04   | WHILE (the constraints are violated)   |
| 05   | Randomly re-generate the current particle $s_t^\lambda$  |
| 06   | END WHILE  |
| 07   | END FOR  |
| 08   | Set $t = 1$  |
| 09   | Generate local best: Set $p_t^\lambda = s_t^\lambda$   |
| 10   | Generate global best: Find $F(s_t^\lambda)$ , $p_t^g$ is set to the position of $s_t^\lambda$  |
| 11   | WHILE (the terminating criterion is not met)   |
| 12   | FOR (each particle(ant) $\lambda$ in the swarm(colony))  |
| 13   | Generate the velocity and update the position of the current particle $s_t^\lambda$  |
| 14   | Constraint-handling:<br>Check whether the current particles violates the problem constraints or not.<br>If it does, reset it to the previous position $s_{t+1}^\lambda$  |
| 15   | Calculate the fitness value $F(s_t^\lambda)$ of the current particle   |
| 16   | Generate the position of the current ant $z_t^\lambda = \mathcal{N}(p_t^g, \sigma)$  |
| 17   | Constraint-handling:<br>Check whether the current ant violates the problem constraints or not.<br>If it does, reset it to the current particle $s_t^\lambda$   |
| 18   | Calculate the fitness value $F(z_t^\lambda)$ of the current ant  |
| 19   | Update current particle position:<br>Compare the fitness value of current ant with current particle.<br>If the $F(z_t^\lambda)$ is better than fitness value of $F(p_t^\lambda)$ , set $F(s_t^\lambda) = F(z_t^\lambda)$ and $s_t^\lambda = z_t^\lambda$ |
| 20   | Update local best: Compare the fitness value of $F(p_t^\lambda)$ , with $F(s_t^\lambda)$ .<br>If $F(s_t^\lambda)$ is better than fitness value of $F(p_t^\lambda)$ , set $p_t^\lambda$ to the current position $s_t^\lambda$                             |
| 21   | END FOR  |
| 22   | Update global best: Find the global best position in the swarm.<br>If $F(s_t^\lambda)$ is better than fitness value of $F(p_t^g)$ ,<br>$p_t^g$ is set to the position of the current particle $s_t^\lambda$  |
| 23   | Set $t = t + 1$  |
| 24   | END WHILE  |

pheromone spot) at the global best solution  $p_t^g$  of the swarm at any iteration  $t + 1$  in each stage of the ant colony algorithm implementation and all ants search for better solutions in the neighborhood of the global best solution [21]. The pseudo-code for the PSO-ACO algorithm is listed in Table 3. This algorithm was programmed in C++, and used to calculate the binary interaction parameters for minimization of the difference between calculated and experimental bubble pressure for each systems used. Fig. 1 shows a flow diagram of the algorithm used for the phase equilibrium modeling.

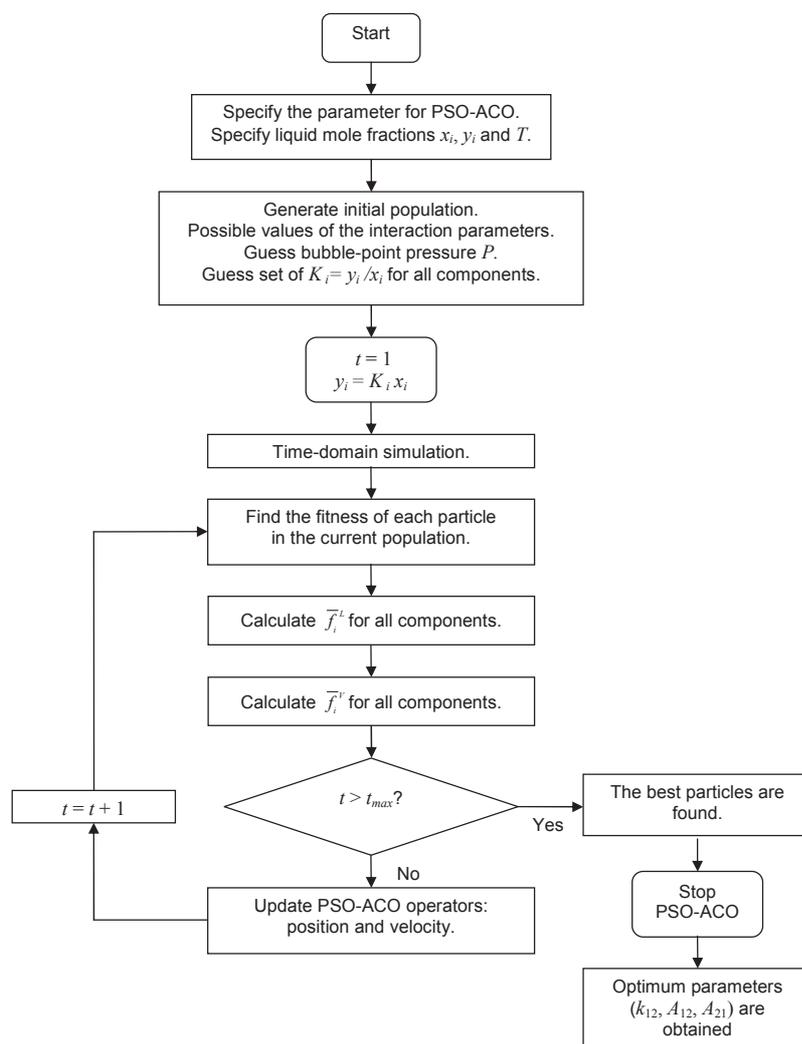


Figure 1: Flow diagram of the total algorithm used for the phase equilibrium modeling.

## 4 Results and discussion

Nine binary vapor-liquid phase systems containing supercritical carbon dioxide and imidazolium-based ionic liquids were considered in this study. The anions: bis(trifluoro-methylsulfonyl)imide ([Tf<sub>2</sub>N]), hexafluorophosphate ([PF<sub>6</sub>]), and tetrafluoroborate ([BF<sub>4</sub>]) are the ones presenting the highest supercritical carbon dioxide solubility. Although both anion and cation influence the carbon dioxide solubility, the anion has the strongest influence [1]. The most common 1-*alkyl*-3-methylimidazolium cations were used, and included: 1-ethyl-3-methylimidazolium ([C<sub>2</sub>mim]), 1-butyl-

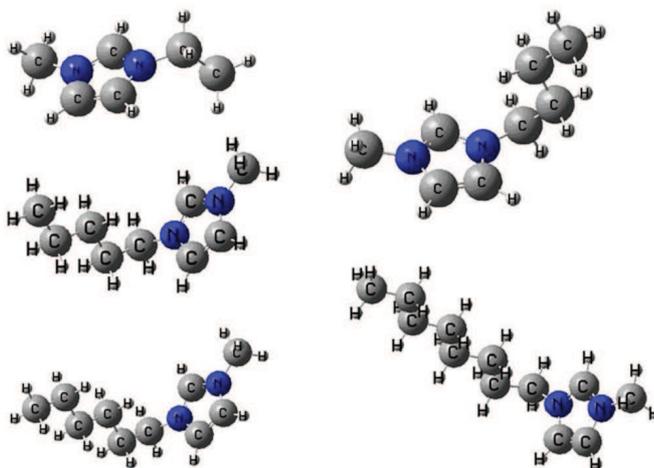


Figure 2: Chemical structures of the series 1-*alkyl*-3-methylimidazolium cation in descending order: ethyl, butyl, pentyl, hexyl, and octyl.

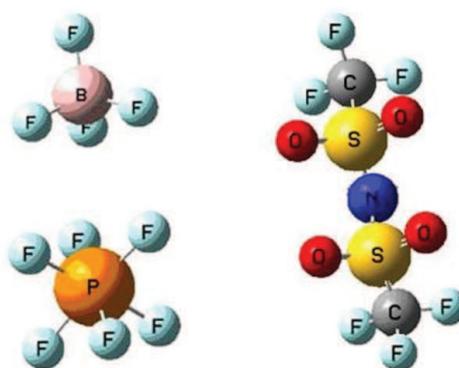


Figure 3: Chemical structures of the anion used: tetrafluoroborate ([BF<sub>4</sub>]), hexafluorophosphate ([PF<sub>6</sub>]), and bis(trifluoromethylsulfonyl)imide ([Tf<sub>2</sub>N]).

3-methylimidazolium ([C<sub>4</sub>mim]), 1-pentyl-3-methylimidazolium ([C<sub>5</sub>mim]), 1-hexyl-3-methylimidazolium ([C<sub>6</sub>mim]), and 1-octyl-3-methylimidazolium ([C<sub>8</sub>mim]). Fig. 2 shows the series 1-*alkyl*-3-methylimidazolium cation (in descending order: ethyl, butyl, pentyl, hexyl, and octyl). Fig. 3 shows the structure of the anions used.

Table 4 shows the thermodynamic properties of the substances involved in the study. In this table,  $T_c$  is the critical temperature,  $P_c$  is the critical pressure, and  $\omega$  is the acentric factor. The data for the ionic liquids were taken from the literature [23,24]. The data of the carbon dioxide were taken from Daubert *et al* [25].

The details of the experimental vapor-liquid equilibrium data taken from references [26–28] are presented in Table 5. As seen in the table, the temperature and pressure ranges are narrow and go from 313K to 333K and from 1 to 13 MPa, respectively.

Table 4: Thermodynamic properties of the substances involved in this study.

| No. | Substance                               | $T_c$ (K) | $P_c$ (MPa) | $\omega$ |
|-----|---|-----------|-------------|----------|
| 1   | [C <sub>2</sub> mim][Tf <sub>2</sub> N] | 1214.2    | 3.37        | 0.2818   |
| 2   | [C <sub>4</sub> mim][Tf <sub>2</sub> N] | 1265.0    | 2.76        | 0.2656   |
| 3   | [C <sub>5</sub> mim][Tf <sub>2</sub> N] | 1249.4    | 2.63        | 0.4123   |
| 4   | [C <sub>6</sub> mim][Tf <sub>2</sub> N] | 1287.3    | 2.39        | 0.3539   |
| 5   | [C <sub>8</sub> mim][Tf <sub>2</sub> N] | 1311.9    | 2.10        | 0.4453   |
| 6   | [C <sub>4</sub> mim][PF <sub>6</sub> ]  | 708.9     | 1.73        | 0.7553   |
| 7   | [C <sub>8</sub> mim][PF <sub>6</sub> ]  | 800.1     | 1.40        | 0.9069   |
| 8   | [C <sub>4</sub> mim][BF <sub>4</sub> ]  | 632.3     | 2.04        | 0.8489   |
| 9   | [C <sub>8</sub> mim][BF <sub>4</sub> ]  | 726.1     | 1.60        | 0.9954   |
| 10  | CO <sub>2</sub>                         | 304.2     | 7.38        | 0.2236   |

Table 5: Details on the phase equilibrium data of the nine systems used in this study (columns: 1 to 7), and binary interaction parameters calculated with the proposed algorithm (columns: 8 to 10).

| No. | CO <sub>2</sub> +                       | Ref. | $T$ (K) | $\Delta P$ (MPa) | $\Delta x$ (CO <sub>2</sub> ) | $N_D$ | $k_{12}$ | $A_{12}$ | $A_{21}$ |
|-----|---|------|---------|------------------|-------------------------------|-------|----------|----------|----------|
| 1   | [C <sub>2</sub> mim][Tf <sub>2</sub> N] | [26] | 323     | 1-11             | 0.2-0.7                       | 9     | 0.0096   | -1.5396  | 4.8771   |
|     |   |      | 333     | 1-11             | 0.2-0.7                       | 9     | 0.0059   | -1.6351  | 4.3253   |
| 2   | [C <sub>4</sub> mim][Tf <sub>2</sub> N] | [27] | 313     | 1-13             | 0.2-0.8                       | 8     | 0.3386   | -0.2101  | -0.6716  |
|     |   |      | 333     | 1-13             | 0.2-0.7                       | 8     | 0.2888   | -1.0198  | -0.4726  |
| 3   | [C <sub>5</sub> mim][Tf <sub>2</sub> N] | [26] | 323     | 1-13             | 0.2-0.7                       | 9     | 0.0353   | -0.1315  | 2.0566   |
|     |   |      | 333     | 1-13             | 0.2-0.7                       | 9     | 0.0149   | -0.3224  | 1.9157   |
| 4   | [C <sub>6</sub> mim][Tf <sub>2</sub> N] | [27] | 313     | 1-12             | 0.2-0.8                       | 6     | 0.2799   | -0.0030  | -0.0160  |
|     |   |      | 333     | 1-11             | 0.2-0.7                       | 8     | 0.1810   | -0.5586  | 0.0241   |
| 5   | [C <sub>8</sub> mim][Tf <sub>2</sub> N] | [27] | 313     | 1-12             | 0.2-0.8                       | 8     | 0.3341   | -0.2594  | -0.6778  |
|     |   |      | 333     | 1-12             | 0.2-0.8                       | 8     | 0.2966   | -0.4801  | -0.5588  |
| 6   | [C <sub>4</sub> mim][PF <sub>6</sub> ]  | [28] | 313     | 1-10             | 0.2-0.7                       | 8     | 0.2304   | 0.6102   | 4.2140   |
|     |   |      | 323     | 1-10             | 0.2-0.7                       | 8     | 0.1629   | 0.4165   | 6.9724   |
|     |   |      | 333     | 1-10             | 0.2-0.7                       | 8     | 0.2968   | 0.2352   | 5.5807   |
| 7   | [C <sub>8</sub> mim][PF <sub>6</sub> ]  | [28] | 313     | 1-10             | 0.2-0.7                       | 8     | 0.1934   | 0.7104   | 5.4292   |
|     |   |      | 323     | 1-10             | 0.2-0.7                       | 8     | 0.1514   | 0.5301   | 6.3123   |
|     |   |      | 333     | 1-10             | 0.2-0.7                       | 8     | 0.2680   | 0.3602   | 5.0220   |
| 8   | [C <sub>4</sub> mim][BF <sub>4</sub> ]  | [27] | 313     | 1-10             | 0.1-0.6                       | 8     | 0.2804   | 0.9204   | 2.4878   |
|     |   |      | 333     | 1-10             | 0.1-0.6                       | 7     | 0.4016   | 0.4432   | 2.9091   |
| 9   | [C <sub>8</sub> mim][PF <sub>4</sub> ]  | [28] | 313     | 1-10             | 0.2-0.7                       | 8     | 0.1798   | 0.7443   | 5.8708   |
|     |   |      | 323     | 1-10             | 0.2-0.7                       | 8     | 0.1465   | 0.5793   | 6.0336   |
|     |   |      | 333     | 1-10             | 0.2-0.7                       | 8     | 0.2542   | 0.4224   | 5.7520   |

Among the many cubic EoS of van der Waals type nowadays available, the proposed by Peng and Robinson (PR-EoS) is widely used because of its simplicity and flexibility [10]. This equation has proven to combine the simplicity and accuracy required for the prediction and correlation of fluid properties, in particular of phase equilibria [4, 6]. The effect of the uncertainty of the critical properties in the phase equilibria calculations

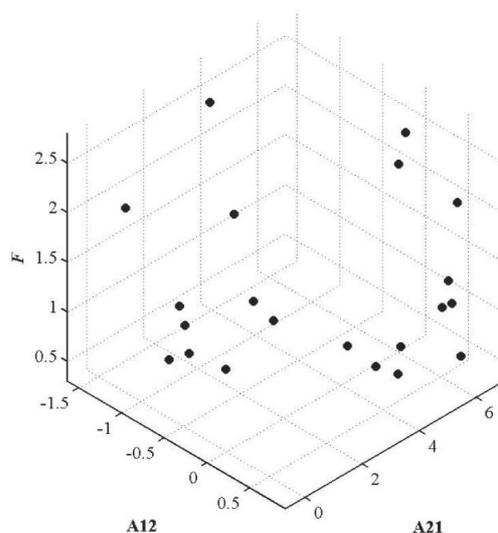


Figure 4: Deviations of the interaction parameters estimated by minimization of the objective function.

using PR-EoS has been investigated for several systems, but the general trend and curvature of the phase equilibrium curve is not altered [29]. The interaction parameters represent the functionality of the constants of the equation with the concentration. It has been recognized that van der Waals mixing rules with one or two parameters don't give good results for systems complex [30]. But, the Wong-Sandler mixing rules have shown to be successful in these applications. In other works to improve the predictions in mixtures, a third interaction parameter has been introduced and has been shown that these mixing rules allow a accurate representation that when the VdW mixing rules are used [1,10,11]. The PR-WS-VL model and the PSO-ACO algorithm were used to calculate  $k_{12}$ ,  $A_{12}$  and  $A_{21}$ , and  $P$  by minimizing the Eq. (2.33), and considering the absolute deviations between experimental and calculated values of bubble point in the vapor-liquid phase of the ionic liquids on the supercritical carbon dioxide. Fig. 4 shows the interaction parameters determined with the hybrid optimization and based on the minimization of the Eq. (2.33). The last three columns in Table 5 show the optimum values calculated for the binary interaction parameters  $k_{12}$ ,  $A_{12}$  and  $A_{21}$  for each system considered in this study. These results show that the pressures of the ionic liquids in the vapor phase were correlated with low deviations between experimental and calculated values (deviations were below than 10%).

Fig. 5 shows the variation of the binary interaction parameters as a function of the absolute temperature. It can be observed the smooth behavior of the parameters included in the PR-WS-VL model. The parameter  $A_{12}$  decreases with the temperature for all cases studied, while the parameter  $A_{21}$  goes through a maximum for all the mixtures. This smooth behavior of the parameters in the Wong-Sandler mixing rules indicates that the model could be better generalized if more systems are studied. Fig. 6 shows an exam-

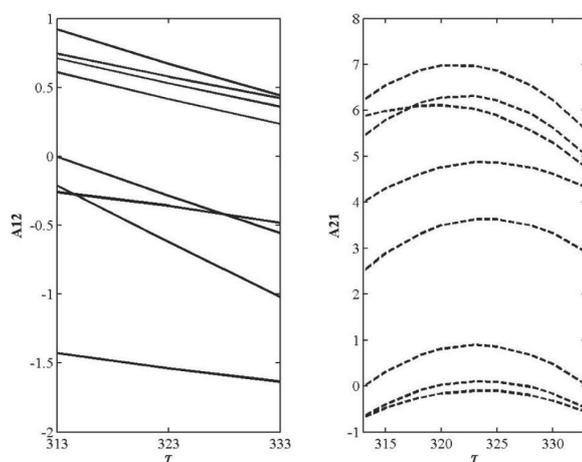


Figure 5: Variation of the binary interaction parameters as a function of the temperature for all systems used.

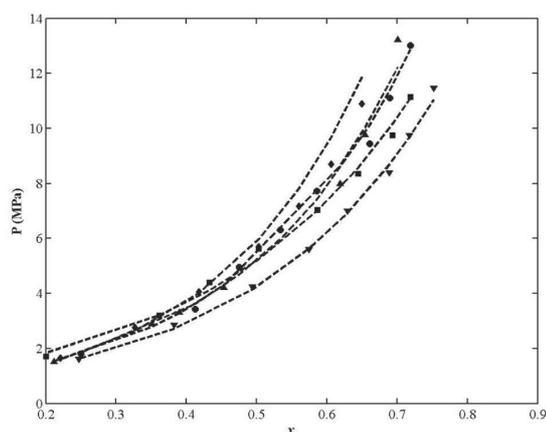


Figure 6: Calculated (---) and experimental (symbols) pressures at 333K for CO<sub>2</sub> with 1-*alkyl*-3-methylimidazolium bis(trifluoromethylsulfonyl)imide: (◆) [C<sub>2</sub>mim], (●) [C<sub>4</sub>mim], (▲) [C<sub>5</sub>mim], (■) [C<sub>6</sub>mim], and (▼) [C<sub>8</sub>mim].

ple of the accuracy of the proposed method to describe the phase equilibrium of binary systems containing carbon dioxide with ionic liquid. This figure shows the pressure-concentration values for all systems studied at 313K. As seen in the picture, the good correlation, represented by the closeness between the experimental data and the correlated values, is observed in all cases. These values calculated with proposed method are believed to be accurate enough for engineering calculations, and for generalized correlations, among other uses.

A comparison was made between of the results obtained with the PSO-ACO algorithm and the results obtained with another two algorithms: genetic algorithm (GA) [31], and Levenberg-Marquart algorithm (LMA) [32]. Note that, GA and LMA are commonly

Table 6: Mean values of the variables of interest for PSO-ACO, GA and LMA.

| Parameter                                   | PSO-ACO | GA    | LMA   |
|---|---------|-------|-------|
| Iteration best solution ( $t_{\max}$ )      | 750     | 350   | 950   |
| CPU time (sec)                              | 560     | 331   | 2074  |
| Unique solution in the final population (%) | 90      | 73    | —     |
| Accuracy of solution (%)                    | 95.31   | 88.74 | 80.22 |
| Minimum deviation (%)                       | 0.31    | 1.52  | 6.03  |
| Maximum deviation (%)                       | 9.14    | 15.05 | 16.27 |
| Average deviation (%)                       | 1.42    | 5.92  | 10.33 |

used in these problems. Table 6 shows the mean values of the above variables of interest for these three algorithms. In this table, the best variables were calculated as an average of the best solution found by the three algorithms for all problems (nine complex mixtures), and to evaluate the quality of the entire set of solutions that each algorithm provides. In general PSO-ACO performs better than GA and LMA, with accuracy of 95% and average deviation below than 2%. Fig. 7 shows a comparison between the PSO-ACO algorithm development in this work, with GA and LMA. This figure shows the average pressure deviations found with the three algorithms for all the ionic liquids considered in this study at 333K. As is observed in the figures, the best method to estimate the vapor-liquid equilibrium of the systems used is the PSO-ACO algorithm.

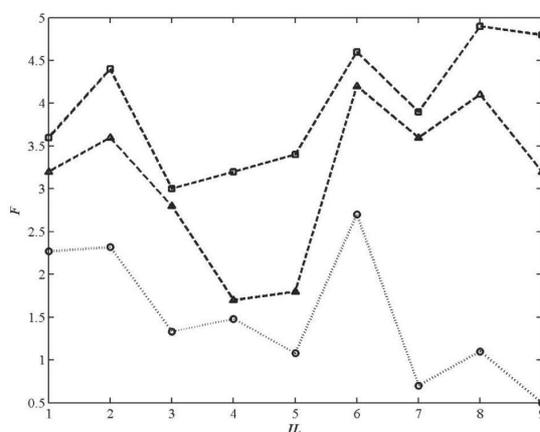


Figure 7: Comparison between PSO-ACO (O), GA ( $\Delta$ ), and LMA ( $\square$ ) optimizations used on all systems of this study at 333K. In this figure, the substances are listed as in Table 5.

Thus, the results show that the implementation of the thermodynamic model PR-WS-VL was crucial, and that the hybrid PSO-ACO algorithm is the preferable method to optimize the interaction parameters of the vapor-liquid equilibrium of binary systems containing carbon dioxide with ionic liquids.

## 5 Conclusion

In this work, nine binary vapor-liquid phase systems containing supercritical carbon dioxide + 1-*alkyl*-3-methylimidazolium ionic liquids were evaluated using a hybrid PSO-ACO algorithm. The Peng-Robinson equation of state was incorporated into the classical equilibrium equation. The mixing rule proposed by Wong-Sandler was used, and the van Laar model was included to evaluate the excess Gibbs free energy that appear in this mixing rule. The PSO-ACO algorithm was used to minimize the difference between calculated and experimental bubble pressure, and the optimal interaction parameters were determined by data regression. Based on the results and discussion presented in this study, the following main conclusions are obtained: i) the thermodynamic PR-WS-VL model is appropriate for modeling the phase equilibrium of binary systems containing supercritical fluid + ionic liquids; ii) the hybrid PSO-ACO algorithm is a good tool to calculate the optimum values for the binary interaction parameters ( $k_{12}$ ,  $A_{12}$ ,  $A_{21}$ ), and phase equilibrium were correlated with low deviations between experimental and calculated values; iii) The values calculated with proposed method are believed to be accurate enough for engineering calculations, and for generalized correlations, among other uses.

## Acknowledgments

The author thank the Direction of Research of the University of La Serena (DIULS), and the Department of Physics of the University of La Serena (DFULS) for the special support that made possible the preparation of this paper.

## References

- [1] J. A. Lazzús, A. A. Pérez Ponce and L. O. Palma Chilla, *Fluid Phase Equilib.* 317, 132 (2012).
- [2] L. A. Blanchard, D. Hancu, E. J. Beckman and J. F. Brennecke, *Nature* 399, 28 (1999).
- [3] G. Hong, J. Jacquemin, M. Deetlefs, C. Hardacre, P. Husson and M. F. Costa Gomes, *Fluid Phase Equilib.* 257, 27 (2007).
- [4] J. A. Lazzús, *J. Eng. Thermophys.* 18, 306 (2009).
- [5] J. A. Lazzús, *Comput. Math. Appl.* 60, 2260 (2010).
- [6] J. A. Lazzús and L. O. Palma Chilla, *J. Eng. Thermophys.* 20, 101 (2011).
- [7] J. Kennedy, R. C. Eberhart and Y. Shi, *Swarm Intelligence* (Morgan Kaufman, San Francisco, 2001).
- [8] M. Dorigo and L. M. Gambardella, *Biosystems* 43, 73 (1997).
- [9] S. M. Walas, *Phase Equilibria in Chemical Engineering* (Butterworth-Heinemann, Boston, 1985).
- [10] H. Orbey and S. I. Sandler, *Modeling Vapor-Liquid Equilibria. Cubic Equations of State and Their Mixing Rules* (Cambridge University Press, USA, 1998).
- [11] J. A. Lazzús and J. Marín, *J. Eng. Thermophys.* 19, 170 (2010).
- [12] D. Y. Peng and D. B. Robinson, *Ind. Eng. Chem. Fund.* 15, 59 (1976).
- [13] D. S. H. Wong and S. I. Sandler, *AIChE J.* 38, 671 (1992).

- [14] Y. Jiang, T. Hu, C. C. Huang and X. Wu, *Appl. Math. Comput.* 193, 231 (2007).
- [15] T. Da and G. Xiurun, *Neurocomputing* 63, 527 (2005).
- [16] Y. Shi and R. Eberhart, A modified particle swarm optimizer IEEE International Conference on Evolutionary Computation (IEEE Press, Piscataway, 1998) pp. 69–73.
- [17] M. S. Kiran, E. Ozceylan, M. Gunduz and T. Paksoy, *Energy Convers. Manage.* 53, 75 (2012).
- [18] T. Niknam and B. Amiri, *Appl. Soft Comput.* 10, 183 (2010).
- [19] R. J. Muller, D. Monekosso, S. Barman and P. Remagnino, *Exp. Syst. Appl.* 36, 9608 (2009).
- [20] P. S. Shelokar, P. Siarry, V. K. Jayaraman and B. D. Kulkarni, *Appl. Math. Comput.* 188, 129 (2007).
- [21] A. Kaveh and S. Talatahari, *Asian J. Civil Eng.* 10, 611 (2009).
- [22] Y. Marinakis, M. Marinaki, M. Doumpos and C. Zopounidis, *Exp. Syst. Appl.* 36, 10604 (2009).
- [23] J. O. Valderrama and P. A. Robles, *Ind. Eng. Chem. Res.* 46, 1338 (2007).
- [24] J. O. Valderrama, W. W. Sanga and J. A. Lazzús, *Ind. Eng. Chem. Res.* 47, 1318 (2008).
- [25] T. E. Daubert, R. P. Danner, H. M. Sibul and C. C. Stebbins, *Physical and Thermodynamic Properties of Pure Chemicals. Data Compilation* (Taylor & Francis, London, 1996).
- [26] P. J. Carvalho, V. H. Alvarez, J. J. B. Machado, J. Pauly, J. L. Daridon, I. M. Marrucho, M. Aznar and J. A. P. Coutinho, *J. Supercrit. Fluids* 48, 99 (2009).
- [27] S. N. V. K. Aki, B. R. Mellein, E. M. Saurer and J. F. Brennecke, *J. Phys. Chem. B* 108, 20355 (2004).
- [28] L. A. Blanchard, Z. Gu and J. F. Brennecke, *J. Phys. Chem. B* 105, 2437 (2001).
- [29] P. Coutsikos, K. Magoulas and G. M. Kontogeorgis, *J. Supercrit. Fluids* 25, 197 (2003).
- [30] M. A. Trebble and P. R. Bishnoi, *Fluid Phase Equilib.* 40, 1 (1988).
- [31] J. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, USA, 1975).
- [32] M. Reilly, *Computer Programs for Chemical Engineering Education* (Sterling Swift, Texas, 1972).