Journal of Molecular Liquids 196 (2014) 167–177

Contents lists available at ScienceDirect

Journal of Molecular Liquids

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

Hybrid swarm optimization for vapor–liquid equilibrium modeling

Juan A. Lazzús *

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

ARTICLE INFO

Article history:
Received 27 January 2014
Received in revised form 21 March 2014
Accepted 23 March 2014
Available online 5 April 2014

Keywords:
Vapor–liquid equilibrium
Peng–Robinson equation
NRTL model
UNIQUAC model
Binary interaction parameters
Particle swarm optimization
Ant colony optimization

ABSTRACT

A hybrid algorithm based on particle swarm optimization and ant colony optimization was used to describe the vapor–liquid equilibrium of complex mixtures. The proposed PSO + ACO algorithm is tested on several benchmark functions from the usual literature. Firstly, nine binary vapor–liquid phase systems containing supercritical fluids and ionic liquids were evaluated for optimizing the equation of state method. Next, twenty binary vapor–liquid phase systems were described using two activity coefficient models optimized by the hybrid algorithm. The results of vapor–liquid equilibrium modeling were compared with the Levenberg–Marquardt algorithm, and show that the application of PSO + ACO algorithm on thermodynamic models such as equation of state methods and activity coefficient models, is crucial, and that the hybrid PSO + ACO algorithm is a good tool to optimize the interaction parameters to describe the vapor–liquid equilibrium of several systems.

1. Introduction

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, nowadays, is represented as an intelligent search problem, where one or more agents are employed to determine the optimum on a search landscape [1]. Modern optimization techniques have aroused great interest among the scientific and technical communities in a wide variety of fields recently, because of their ability to solve problems with non-linear and non-convex dependence of design parameters [2].

The accurate prediction of physical properties on phase equilibrium can be considered one of the most important applications in thermodynamic processes [3]. The most common way to face this task is to fit the experimental data to a thermodynamic model and use the obtained model with fitted parameters for predicting properties at other conditions. The existing methods to solve phase equilibrium systems obtain only local solutions. It has been demonstrated that for several cases, 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 for the development of mathematical models obtained from experimental data [5].

Thus, the use of heuristic optimization methods, such as particle swarm optimization [6] and ant colony optimization [7], for the parameter estimation is very promising [4]. These biologically-deriver methods represent an excellent alternative to find a global optimum for phase equilibrium calculations [5].

In this work, a hybrid algorithm based on particle swarm optimization and ant colony optimization was used to describe the vapor–liquid equilibrium of complex mixtures. Firstly, nine binary vapor–liquid phase systems containing supercritical fluids and ionic liquids were evaluated for optimizing the equation of state method. Next, twenty binary vapor–liquid phase systems were described using two activity coefficient models optimized by the hybrid algorithm. Both methods were used to calculate the binary interaction parameters of these mixtures by the minimization of the difference between calculated and experimental data.

2. Optimization method

The hybrid algorithm was developed with the particle swarm optimization and ant colony optimization. Particle swarm optimization is one of the recent meta-heuristic techniques proposed by Kennedy and Eberhart [6]. Ant colony optimization is an algorithm based on the foraging behavior of ants, and has been first introduced by Dorigo [7].

2.1. 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 [6].

* Tel.: +56 51 204128; fax: +56 51 206658.
E-mail address: jlazzus@dfuls.cl.

http://dx.doi.org/10.1016/j.molliq.2014.03.031
0167-7322/

The particle swarm algorithm is initialized with a population of random particles and the algorithm searches for optima by updating generations [8]. 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 [9]. The performance of each particle is evaluated using a predefined fitness function, which encapsulates the characteristics of the optimization problem [10].

Let s and v denote a particle position and its corresponding velocity in a search space, respectively [6]. Therefore, the $\lambda - th$ particle is represented in the n-dimensional search space as [1]:

$$s^{\lambda} = \left(s_1^{\lambda}, s_2^{\lambda}, \cdots, s_n^{\lambda}\right). \tag{1}$$

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

$$v^{\lambda} = \left(v_1^{\lambda}, v_2^{\lambda}, \cdots, v_n^{\lambda}\right). \tag{2}$$

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

$$p^{\lambda} = \left(p_1^{\lambda}, p_2^{\lambda}, \cdots, p_n^{\lambda}\right). \tag{3}$$

Then the best position p^{λ} is determined by:

$$p_{t+1}^{\lambda} = \begin{cases} p_t^{\lambda} & \text{if } F\left(s_{t+1}^{\lambda}\right) \geq F\left(p_t^{\lambda}\right) \\ s_{t+1}^{\lambda} & \text{if } F\left(s_{t+1}^{\lambda}\right) < F\left(p_t^{\lambda}\right) \end{cases}. \tag{4}$$

Let t be a time instant. The new particle position is computed by adding the velocity vector to the current position [6]:

$$s_{t+1}^{\lambda} = s_t^{\lambda} + v_{t+1}^{\lambda} \tag{5}$$

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

The velocity update equation is given by:

$$v_{t+1}^{\lambda} = w_t v_t^{\lambda} + c_1 r_1 \left(p_t^{\lambda} - s_t^{\lambda}\right) + c_2 r_2 \left(p_t^{g} - s_t^{\lambda}\right) \tag{6}$$

where w is the inertia weight, c_1 and c_2 are the acceleration constants, and r_1 and r_2 are the elements from two random sequences in the range (0,1). The current position of the particle is determined by s_t^{λ}; p_t^{λ} is the best one of the solutions that this particle has reached and is the best one of the all solutions that the particles have reached [9].

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 [11]. 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 [9]. The following weighting function is used in Eq. (6):

$$w = w_{max} - \frac{w_{max} - w_{min}}{t_{max}} t \tag{7}$$

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 in the range $[-v_{max}, v_{max}]$ to control the excessive roaming of particles outside the search space [8,10]. 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 [9].

2.2. Ant colony optimization

The basic idea of ant colony optimization is to imitate the cooperative behavior of ant colonies [7]. As soon as an ant finds a food source, it evaluates it and carries some food back to the nest [12].

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

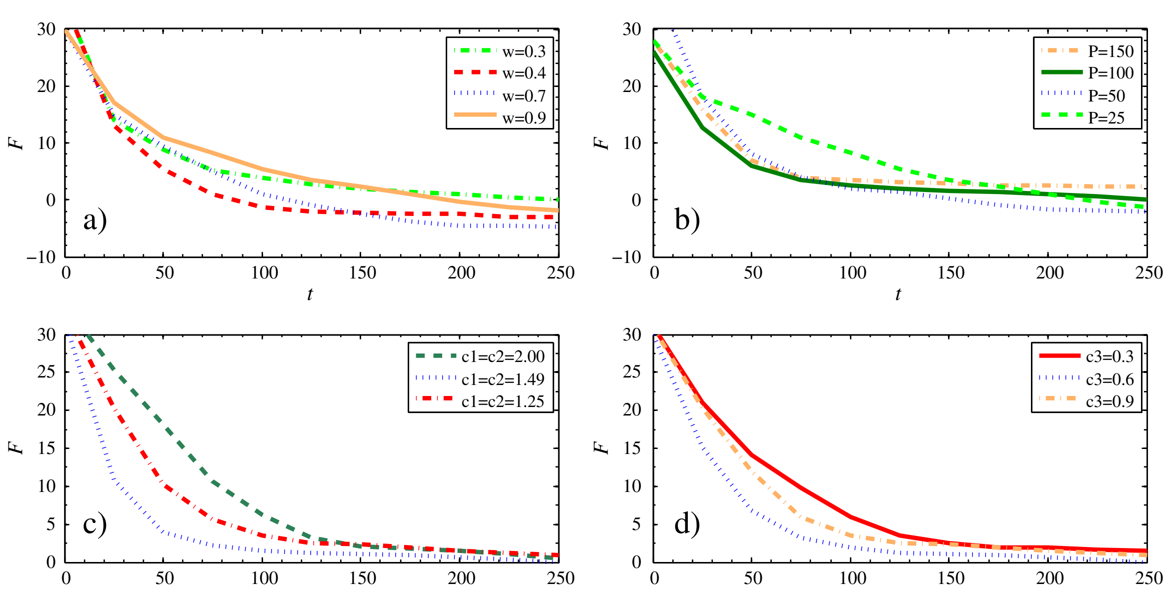

Fig. 1. Convergence graphics. Determination of the best values for: (*a*) inertia weight w; (*b*) population P; (*c*) constants c_1 and c_2; (*d*) constant c_3.