On the Evaluation of Interfacial Tension (IFT) of Paraffins-CO_2 for Enhanced Oil Recovery Process: Comparison of Empirical Correlations, Hybrid Soft Computing Approaches, and Parachor Model

Farzaneh Rezaei [1], Amin Rezaei [2], Saeed Jafari [1], Abdolhossein Hemmati-Sarapardeh [1], *, Amir H. Mohammadi [3], Sohrab Zendehboudi [4]

Supplementary File

S1. Optimization Techniques

Levenberg-Marquardt (LM). Sometimes referred to as the damped LSM, the Levenberg-Marquardt (LM) algorithm has been commonly applied to obtain optimal values of the weights and biases of the multilayer perceptron (MLP) neural networks and find solutions for nonlinear least-square problems. Although the LM is known for a local minimum, rather than a global one, usually it is able to provide a solution even where the starting point is very far from the solution. The form of the performance function of the LM algorithm (sum of squares) eliminates the need for evaluating the Hessian matrix. Indeed, in such cases, the matrix is approximated by computing the gradient as follows [1,2]:

$$H = J^T J \tag{S1}$$

$$g = J^T e \tag{S2}$$

in which, J defines a Jacobian matrix of the first-order derivatives of the network errors with respect to the biases and weights; and e introduces the vector of the network errors. The following Newton-like updating scheme is then applied with the Hessian matrix:

$$x_{k+1} = x_k - (J^T J - \delta I)^{-1} J^T e \quad \text{(S3)}$$

where x refers to the connection weight; and δ is a constant value. The value of δ is declined over successive iterations, with exceptional increases when the performance function is enhanced upon a tentative step. Details on the use of the LM as a training algorithm for NNs are given in the literature [1].

Genetic Algorithm (GA). GA is a powerful evolutionary technique for addressing complicated optimization problems having any type of objective function. First, an initial generation of solutions is represented by a population of randomly generated chromosomes. Subsequently, GA operators, including the crossover, reproduction, and mutation, are applied to evolve the initial chromosomes in an attempt to approach the optimum solution. In the framework of GA, the offspring production probability is measured by two parameters, namely the crossover factor (CF) and the mutation factor (MF), which refer to the probabilities of a binary section of the chromosomes changing from zero to one or vice versa, respectively [3–5]. The GA begins with generating an initial population of chromosomes (solution) and setting the MF and CF. It then evaluates a fitness value for each chromosome as $F_i^k = f\left(X_i^k\right), \forall i$ and an index of the best chromosome as b = {1, 2, …, N}. The next stage is to apply the selection, crossover, and mutation operators to produce a new generation of solutions $X_i^{k+1}, \ \forall i$, followed by re-evaluating the fitness values for the new solutions, $F_i^{k+1} = f\left(X_i^{k+1}\right)$, and finding the best chromosome, b_1. Accordingly, if the fitness value is any better (smaller) than that of the previous generation, the new one replaces

the old one. This procedure is iterated until either the maximum number of iterations or the stopping criterion is met.

Gravitational Search Algorithm (GSA). GSA is a special form of the so-called conjugate gradient method where the agents are considered as objects with their masses measuring their performance. Depending on their masses (i.e., the gravitational force), the objects tend to attract one another, leading to a universal movement towards the heavier objects. Representing better solutions, the heavier masses move at slower velocities than the lighter ones, ensuring the exploitation step of the algorithm. As far as the GSA is concerned, a mass is characterized by four characteristics, including its position and inertial mass as well as passive and active gravitational mass [6]. The mass position is indeed a possible solution of the problem, with the inertial and gravitational masses evaluated through a fitness function. Therefore, the GSA resembles an independent network of masses or a miniaturized artificial world of masses interacting, according to Newton's laws of gravity and motion, as follows:

- Law of gravity: there is an attraction (gravitational) force between any pair of massy particles, with the force being, directly and indirectly, proportional to their masses and distance, respectively.
- Law of motion: the current velocity of a mass can be evaluated as its previous velocity plus/minus some velocity change. Accordingly, the velocity change (i.e., the acceleration) of a mass can be calculated as the force exerted onto the system divided by the inertial mass.

Imperialistic Competitive Algorithm (ICA). ICA is a novel GA-derived approach to optimization. Following the rules of social communities, the ICA was first introduced by Atashpaz-Gargari and Lucas, who highlighted its excellent efficiency in finding the global optima [7,8]. Referring to ICA, three terms have been commonly used: (1) cost function, which serves as a basis for the optimization process, (2) decade, which refers to a single iteration through the entire process of the ICA algorithm, and (3) country, which holds the same concept as the chromosome in the GA. Evaluating the cost function, the country for which the cost value is minimal represents an imperialist, with the rest of countries regulated as colonies [9]. The ICA works based on three essential operators: (1) competition, which makes it possible to control and even acquire colonies of other empires, (2) revolution, which is applied to relocate the countries if a cost value lower than that of the respective imperialist is found, in an attempt to avoid local minima and hence increase the chances of achieving the finest solution, and (3) assimilation, which follows a sociopolitical axis and pushes the colonies toward the imperialist. Based on the value of the total cost function, the competition operator is defined by the following equation [7,9]:

$$TC_n = Cost(imprialist_n) + \eta mean\{Cost(colonies\ of\ empire_n)\} \quad \text{(S4)}$$

where TC and η denote the total cost value and contribution coefficient of the colonies in TC, respectively. Equation (S5) gives the normalization of Equation (S4) as follows [8,9]:

$$NTC_n = TC_n - \max(TC_i) \quad \text{(S5)}$$

In Equation (S5), NTC is termed as the normalized TC. Next, for each empire, the possession probability is calculated by the following relationship:

$$PP_n = \left| \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \right| \quad \text{(S6)}$$

where PP_n and N_{imp} symbolize the possession probability and the imperialist size, respectively. Similar to the GA, the ICA conducts the empire selection; but this approach relies on a single probability value and there is no need for evaluating the cumulative distribution function. This is the main reason that the ICA selection technique is superior over conventional selection techniques, such as the roulette wheel method. In this respect, the following probability vector (P) can be introduced [8,9]:

$$P = [PP_i]_{1\times N_{imp}} \quad \text{(S7)}$$

One should proceed to form a vector of random values and an integrated vector, as follows [8,9]:

$$R = [r_i]_{1\times N_{imp}} \quad \text{(S8)}$$

$$D = P - R = [(pp - r)_i]_{1\times N_{imp}} \quad \text{(S9)}$$

Presently, our goal is to maximize the related indices in vector D for the colonies prepared in each empire. ICA procedural optimization will be terminated when the stopping criterion is accomplished [7].

Particle Swarm Optimization (PSO). PSO is a heuristic approach to the optimization that is inspired by the collective behaviors exhibited by animals such as birds, fishes, and insects [10–12]. Here, a swarm of particles is used to represent a set of probable solutions, and the swarm is

iteratively updated to approach the optimal solution [13]. Accordingly, the terms particle and swarm in the PSO are equivalent to the terms chromosome and population in the GA, respectively [14]. The PSO differs from the GA so that it involves no evolutionary operator (e.g., crossover and mutation). In fact, PSO operates through the particles movements in the search space regulated by the currently optimal particles. In this technique, any particle is affected by the degree to which its topologically neighboring particles are deemed successful [13]. There are various kinds of the topological neighborhood, mainly including the physical and social neighborhoods and the queen [15]. More information concerning PSO can be found in the literature [16].

In a swarm, a particle is identified by its position (vector $x_i(t)$) and velocity (vector $v_i(t)$) that is updated as follows (written for the ith particle) [14,17,18]:

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 \left(p_{best,id}(t) - x_{iid}(t)\right) + c_2 r_2 \left(g_{best,d}(t) - x_{id}(t)\right), \quad \text{(S10)}$$

$$d = 1, 2, \ldots, D$$

In Equation (S10), $p_{best,id}$ resembles the best previous position of particle i; $g_{best,d}$ denotes the best global position; w is the inertia weight, c_1 and c_2 are the learning rates,; and r_1 and r_2 represent the random numbers between 0 and 1 [19]. Equation (S10) is mainly composed of three terms, namely social, cognitive, and inertia components [10,14,20]. The social component (including c_2) measures the fitness value of a particle with reference to its neighboring particles, leading to the swarm throughout the search space accordingly. The cognitive term (including c_1) sets the particle velocities in such a way to push them toward the previously best position. Denoted by $wv_{id}(t)$, the inertia component recalls the previous direction of movement and attempts to keep the particle on the same path at iteration *t*. Overall, these components define a path through which each particle

moves in the search space, where each particle is assigned a new position at each iteration by changing its previous position based on its velocity, as follows:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad \text{(S11)}$$

d = 1, 2, …, D

Ant Colony Optimization (ACO). Formulated by Dorigo, ACO is a population-oriented methodology developed based on the ants' natural tendency toward finding the shortest route connecting their nest to food resources [21–23]. Once such paths (i.e., solutions) were identified, the ants mark them with pheromone to guide the entire population toward establishing a network of the most promising paths (solutions). Since the pheromone-based technique is initially valid for discrete domains, one needs to generalize it before it can be applied to continuous areas. This can be conducted by preparing a set of candidate solutions called a solution archive to build a so-called probabilistic Gaussian mixture model. The solution archive constantly stores a certain number of best solutions that are identified by the ACO. The set of best solutions is then improved by selecting the samples corresponding to the promising solutions out of the probabilistic model [24].

The version of the ACO that has been adapted to continuous domains is called the estimation of distribution algorithm (EDA); the name stems from the probabilistic model-solution archive evolutionary interactions [25–27]. The ACO optimization searches for the lowest value of the cost function by finding a vector x, $x \in X \subseteq R^{n_x}$. For this purpose, one should begin with an initialization stage where *N* solutions are randomly generated and stored in X followed by evaluating their cost values using the cost function. Subsequently, the initial solutions are ranked by their cost values and denoted by x_1 (the best solution) through x_N (the worst solution). As a next

step, each solution in the solution archive is given a weighting factor according to the following expression [24–26]:

$$u_i \propto \frac{1}{\sqrt{2\pi}\alpha N} \exp\left(-\frac{1}{2}\left(\frac{i-1}{\alpha N}\right)^2\right) \tag{S12}$$

Subjected to the following condition:

$$\sum_{i=1}^{N} u_i = 1 \tag{S13}$$

Equation (S13) is then applied to build the probabilistic model in the form of a Gaussian mixture model:

$$G^j(x[j]) = \sum_{i=1}^{N} u_i N(x[j];\ \mu_i[j], \sigma_i[j]\ ) \tag{S14}$$

where

$$N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{S15}$$

where j resembles the decision variable and $x[j]$ indicates the jth element in x.

The model can be further completed by calculating its mean value and standard deviation through the following formulas:

$$\mu_i[j] = x_i[j] \tag{S16}$$

$$\sigma_i[j] = \frac{\xi}{N-1}\sum_{i=1}^{N}|x_i[j] - x_{i'}[j]| \tag{S17}$$

in which, $\xi > 0$ determines the exploitation/exploration balance.

In the next stage, *M* new offspring samples are generated, using a multidimensional model $g = (G^1, G^2, G^{n_x})$, and added to the solution archive followed by assessing the objective function for the newly added samples. Through the next step called selection, a new solution archive is built by selecting *M* recent offsprings and *n* best solutions, where the best solutions represent the optimal solutions. Finally, the stopping criterion is checked, and the optimization process is terminated upon reaching the stopping criterion, with the last best solution acknowledged as the optimization result. The process is iterated if the stopping criterion is not met. Panels (a-e) of Figure S1 present the flowcharts of the GA, GSA, ICA, PSO, and ACO algorithms to implement the RBF modeling.

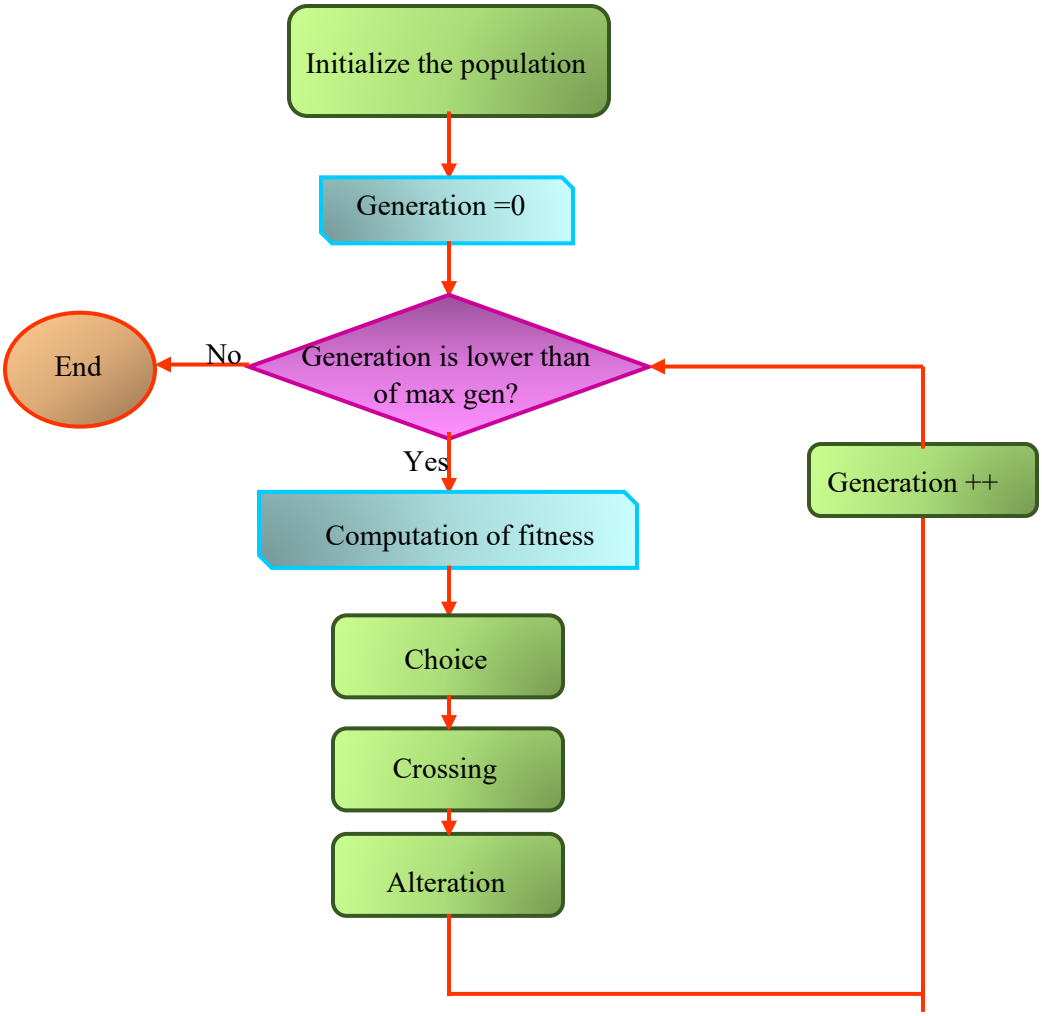


Figure S1a. Flow chart of the GA algorithm used in this study.

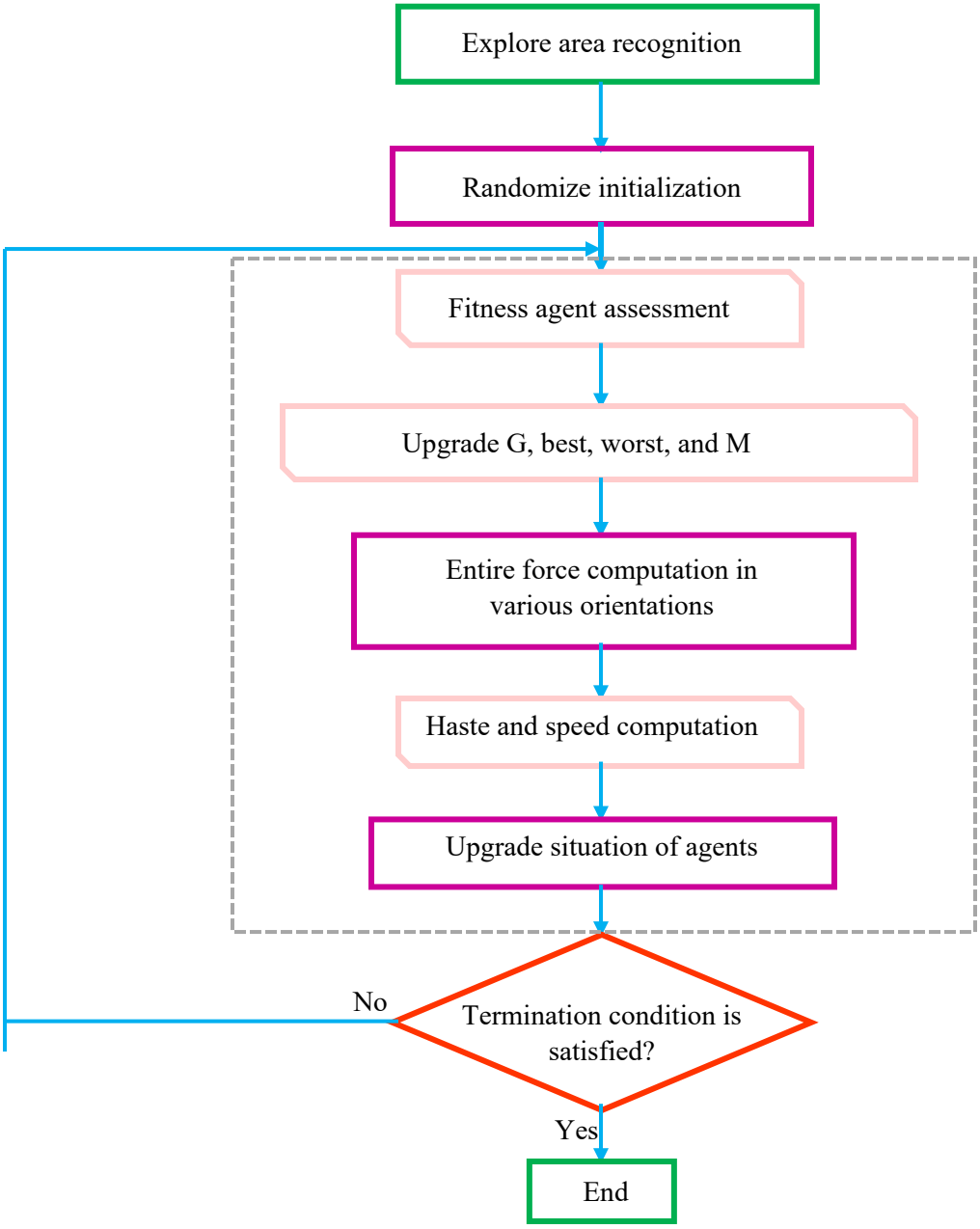


Figure S1b. Flow chart of the GSA algorithm employed in this research.

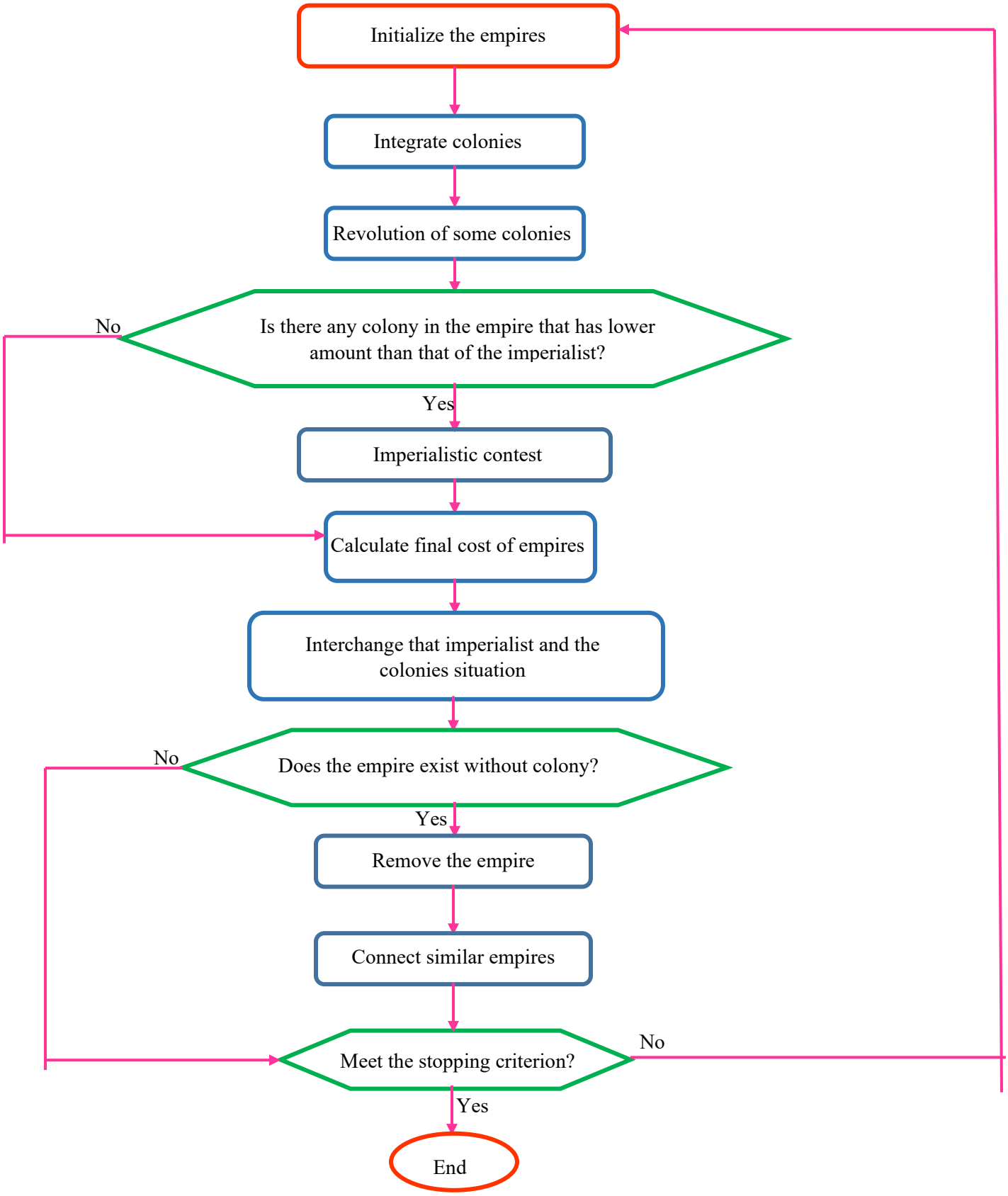


Figure S1c. Main steps of the ICA algorithm used in this research work.

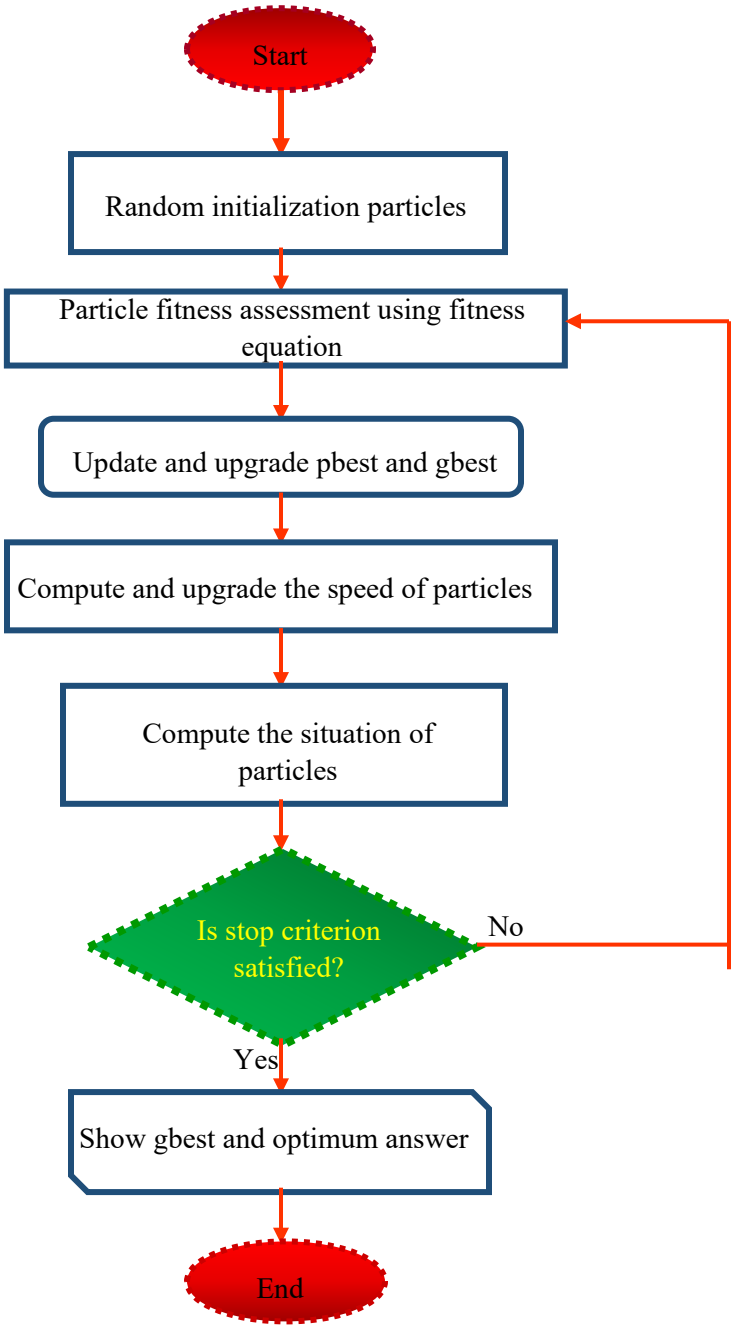


Figure S1d. The flow chart of the PSO algorithm employed in this study.

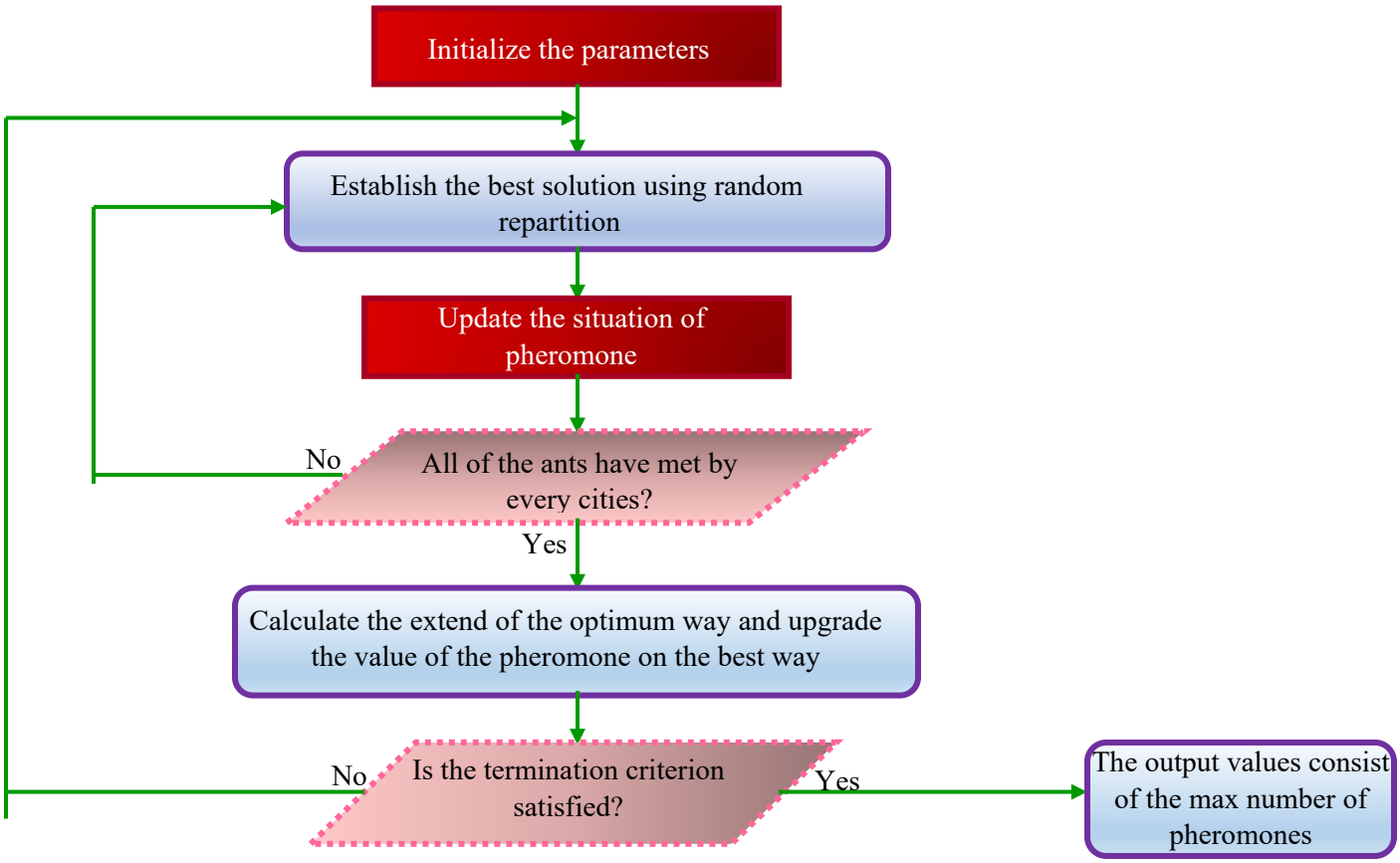


Figure S1e. The flow chart of the ACO algorithm used in this study.

Table S1(a, b, and c) . Developed correlations by the GMDH model to estimate the IFT between CO_2 and n-alkanes with different ranges of molecular weights of n-alkanes.

a) MW of n-Alkane < 128 g.mole^{-1}

N_7	$26.8294 - P \times 7.48134 + P \times T \times 0.0140013 + P^2 \times 0.147394$ $-T^2 \times 0.000101526$
N_6	$-9.98585 + MW \times 0.487082 - MW^2 \times 0.0024677 - P \times 1.99407$ $+P^2 \times 0.0766877$
N_5	$-28.145 + T \times 0.124901 - T \times N_6 \times 0.00856554 - T^2 \times 0.000123$ $+N_6 \times 3.88358 + N_6^2 \times 0.00707468$
N_4	$-18.4425 + MW \times 0.38049 + MW \times N_7 \times 0.00396706$ $-MW^2 \times 0.00193676 + N_7 \times 0.757916 - N_7^2 \times 0.0102219$
N_3	$0.0024115 + N_4 \times 2.10961 + N_4 \times N_5 \times 1.63255 - N_4^2 \times 0.93485$ $-N_5 \times 1.04462 - N_5^2 \times 0.701968$
N_2	$-0.228756 - N_7 \times N_3 \times 0.121174 + N_7^2 \times 0.0455166 + N_3 \times 1.02621$ $+N_3^2 \times 0.0737283$
N_1	$-0.00807249 - N_4 \times N_2 \times 0.557415 + N_4^2 \times 0.26677 + N_2 \times 0.993729 + N_2^2$ $\times 0.289815$
IFT	$2.07822 - T \times 0.00603416 + T \times N_1 \times 0.000913261 + N_1 \times 0.67174$ $+N_1^2 \times 0.0016223$

b) 128 g.mole^{-1} < MW of n-Alkane < 170 g.mole^{-1}

N7	$479.704 - MW \times 5.23017 + MW^2 \times 0.0180574 - T \times 0.481037$ $+T^2 \times 0.000619777$
N6	$27.1628 - P \times 4.0094 + P \times T \times 0.00674831 + P^2 \times 0.0173212$ $-T^2 \times 6.16045 \times 10^{-5}$
N5	$372.423 - MW \times 4.83892 + MW^2 \times 0.016575 - P \times 2.01438 + P^2 \times 0.0508244$

N4	$0.00560628 - N_5 \times N_6 \times 0.232285 + N_5^2 \times 0.136166 + N_6 \times 0.892958$ $+N_6^2 \times 0.102019$
N3	$-18.3745 + T \times 0.0505197 - T \times N_5 \times 0.00461681 + N_5 \times 2.65949$
N2	$-2.87073 + N_7 \times 0.674778 + N_7 \times N_3 \times 0.0212069 - N_7^2 \times 0.0441496$ $+N_3 \times 0.940644 - N_3^2 \times 0.00706968$
N1	$-3.01493 + P \times 0.72071 - P \times N_2 \times 0.0232241 - P^2 \times 0.0366532 + N_2 \times 1.14716$
IFT	$0.00756756 - N_4 \times 0.0484979 + N_1 \times 1.04775$

c) MW of n-Alkane > 170 g.mole^{-1}

N6	$18.8364 - MW \times 0.0221142 - MW \times ln\sqrt{P} \times 0.0205768 + MW^2 \times 0.000126092$ $-(ln\sqrt{P})^2 \times 5.95288$
N5	$-2003.44 - ln\sqrt{P} \times 142.142 + ln\sqrt{P} \times ln\sqrt{T} \times 46.7788 - (ln\sqrt{P})^2 \times 5.72041$ $+ln\sqrt{T} \times 1415.91 - (ln\sqrt{T})^2 \times 247.465$
N4	$-2958.91 + ln\sqrt{T} \times 1939.38 - ln\sqrt{T} \times N_6 \times 6.20335 - (ln\sqrt{T})^2 \times 317.554$ $+N_6 \times 19.7136 - N_6^2 \times 0.0202$
N3	$-1452.61 + ln\sqrt{T} \times 936.662 - ln\sqrt{T} \times N_5 \times 4.50199 - (ln\sqrt{T})^2 \times 150.864$ $+N_5 \times 14.8683 - N_5^2 \times 0.0280386$
N2	$3.8547 - MW \times 0.0496642 + MW^2 \times 0.000147215 + N_3 \times 0.976768$ $+N_3^2 \times 0.000992048$
N1	$5.57983 - ln\sqrt{P} \times 6.01913 + ln\sqrt{P} \times N_2 \times 0.326973 + (ln\sqrt{P})^2 \times 1.59601$ $+N_2 \times 0.386481 + N_2^2 \times 0.0162761$
IFT	$0.152705 - N_4 \times 0.784481 + N_4 \times N_1 \times 0.211075 - N_4^2 \times 0.0960804 + N_1$ $\times 1.76233 - N1^2 \times 0.114078$

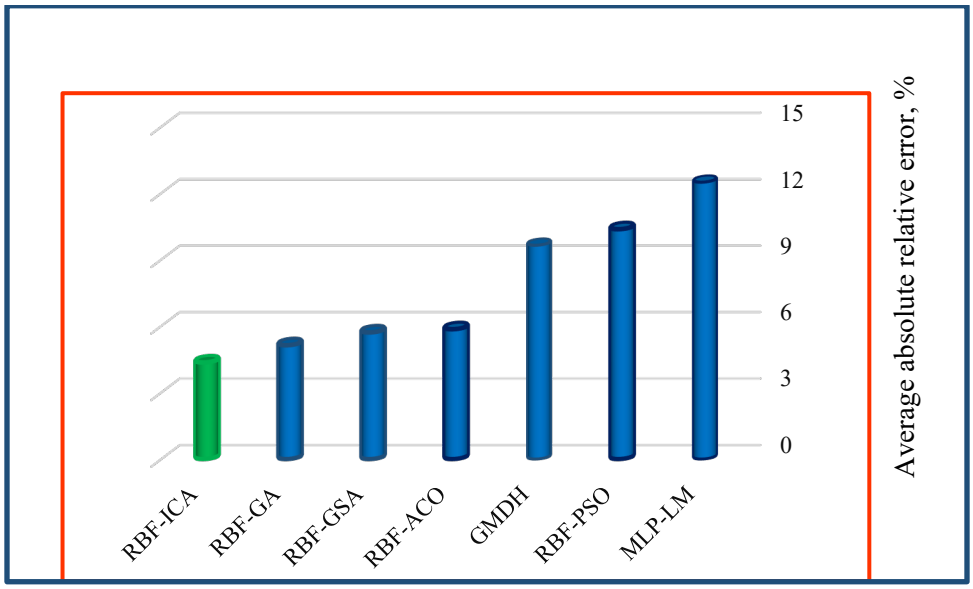
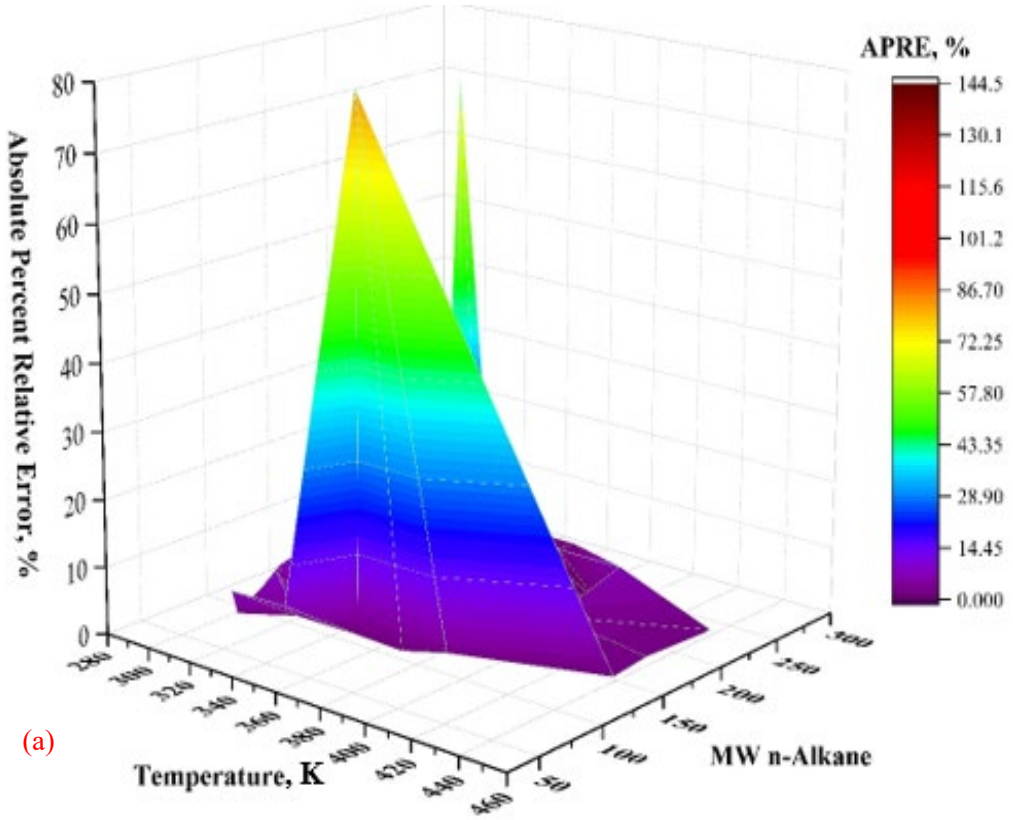
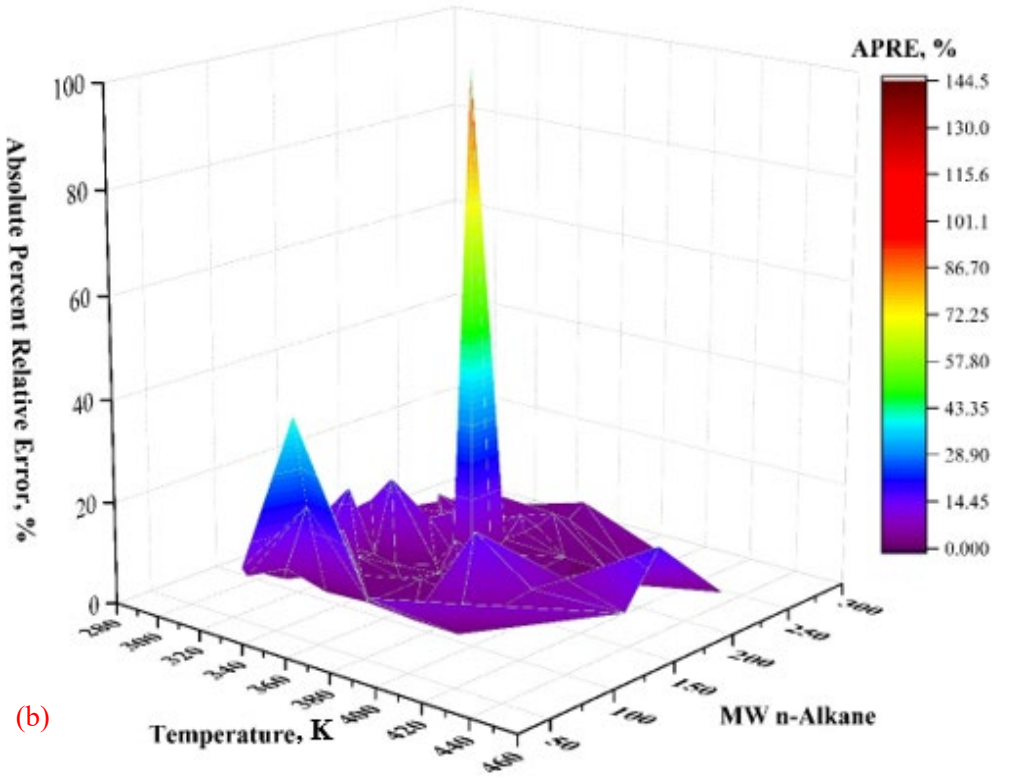


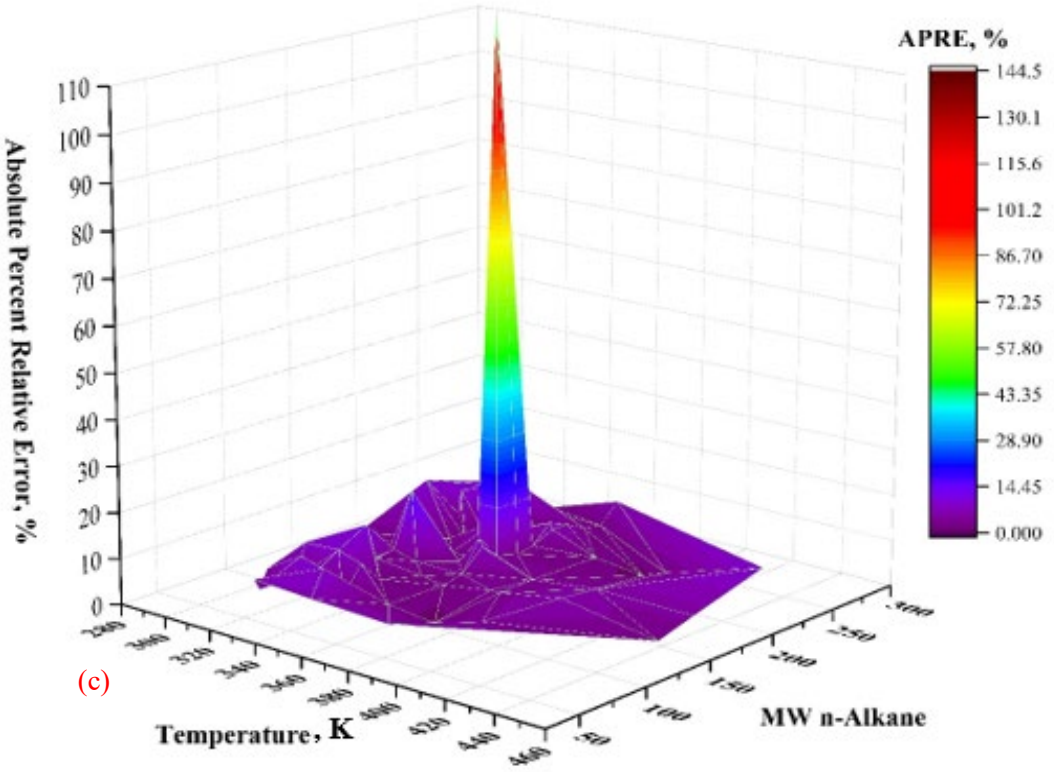
Figure S2. The average absolute relative error based on the models introduced in this study.


APRE, %
144.5
130.1
115.6
101.2
86.70
72.25
57.80
43.35
28.90
14.45
0.000
Absolute Percent Relative Error, %
0
10
20
30
40
50
60
70
80
Temperature, K
MW n-Alkane

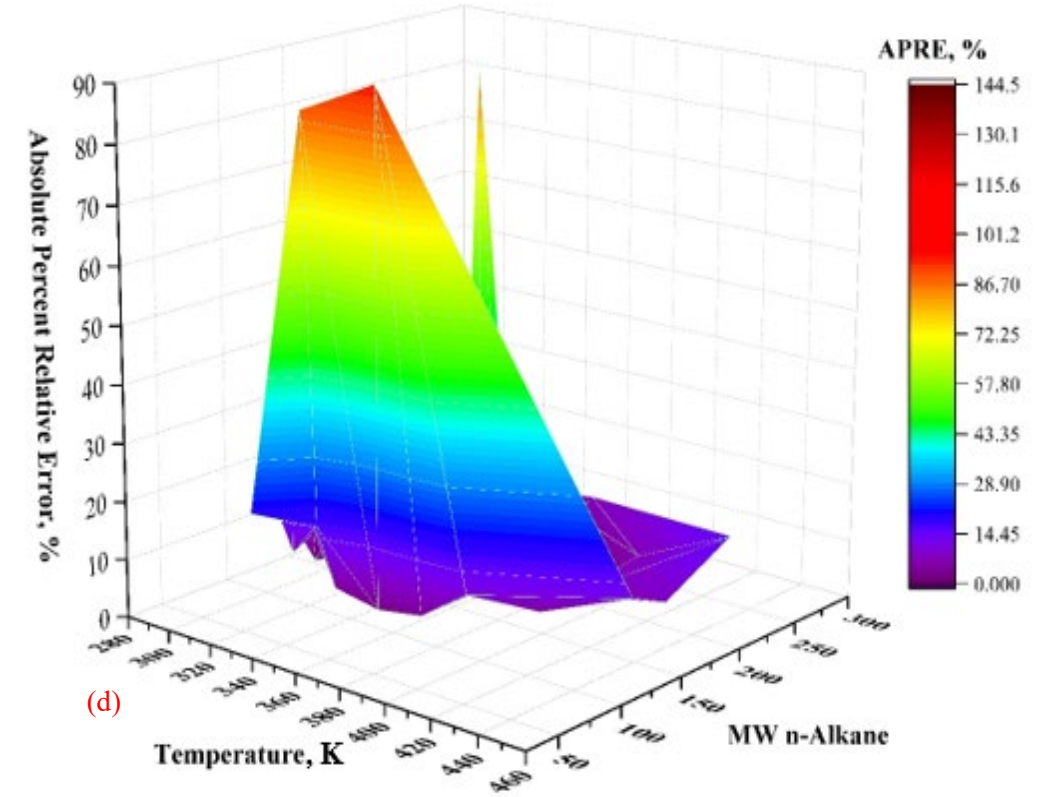
(a)


APRE, %
144.5
130.0
115.6
101.1
86.70
72.25
57.80
43.35
28.90
14.45
0.000
Absolute Percent Relative Error, %
0
20
40
60
80
100
Temperature, K
MW n-Alkane

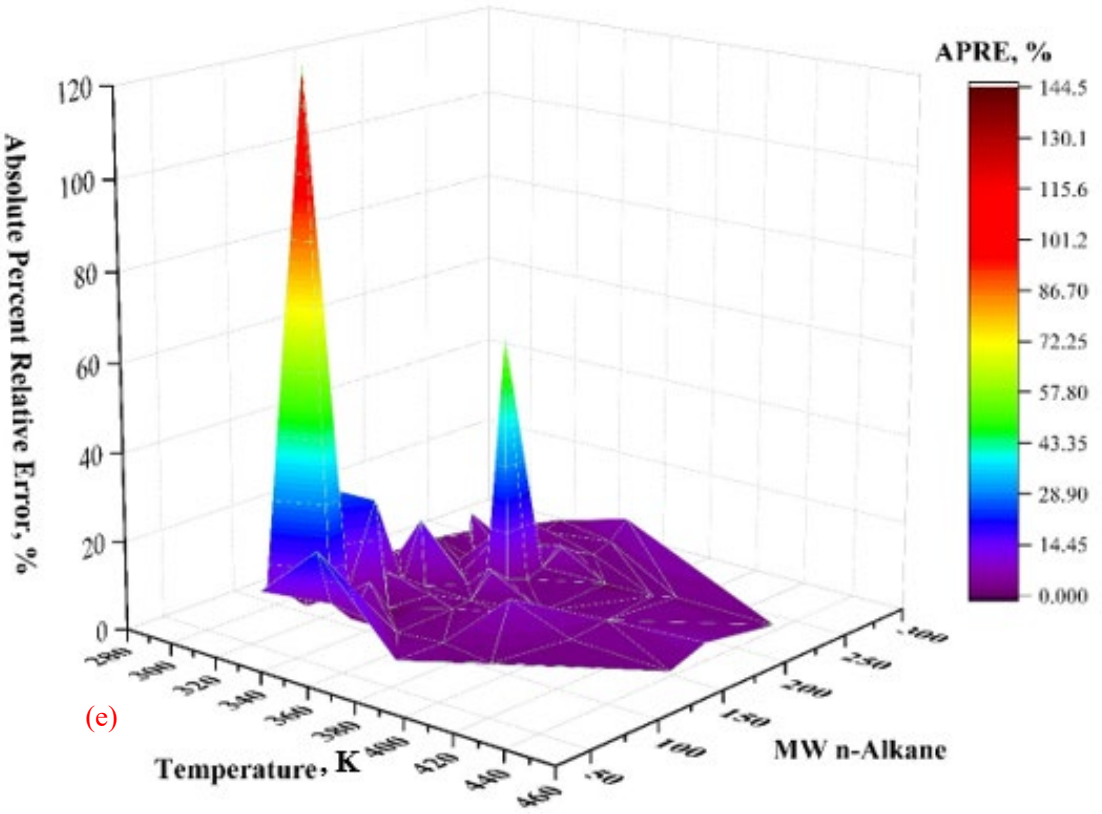
(b)


APRE, %
144.5
130.1
115.6
101.2
86.70
72.25
57.80
43.35
28.90
14.45
0.000
Absolute Percent Relative Error, %
110
100
90
80
70
60
50
40
30
20
10
0
Temperature, K
MW n-Alkane

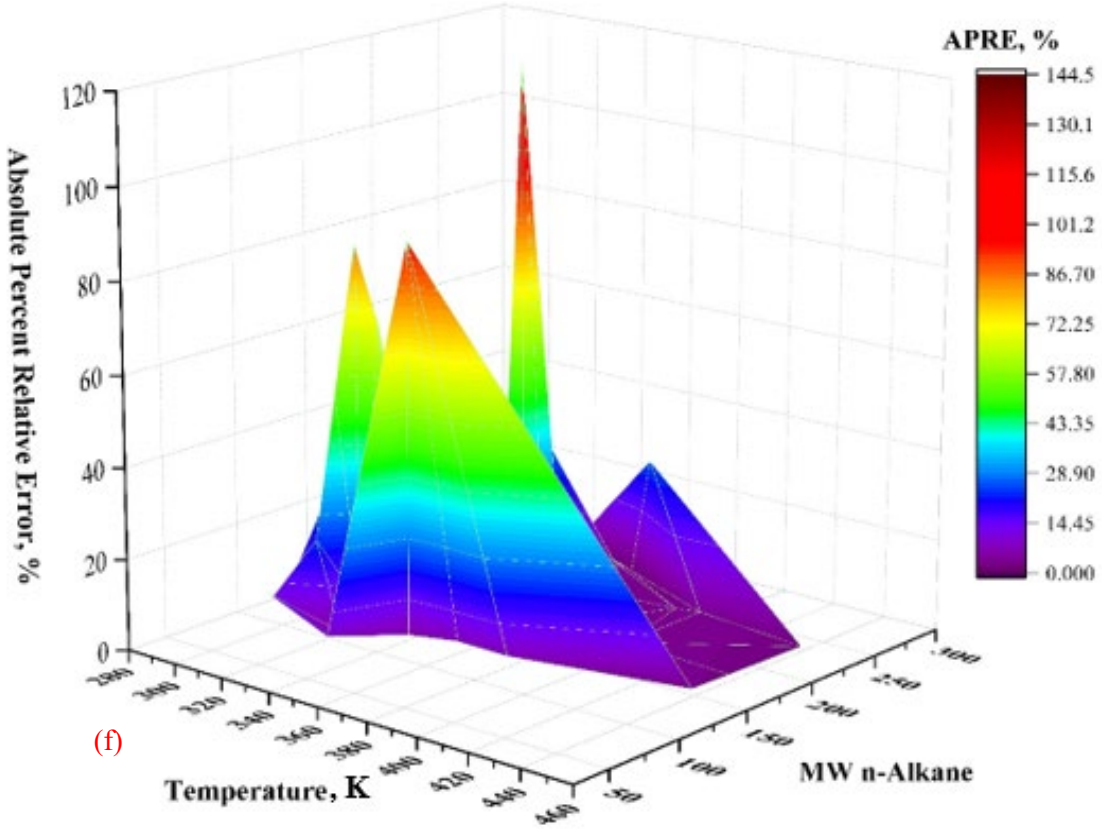
(c)


APRE, %
144.5
130.1
115.6
101.2
86.70
72.25
57.80
43.35
28.90
14.45
0.000
Absolute Percent Relative Error, %
90
80
70
60
50
40
30
20
10
0
Temperature, K
MW n-Alkane

(d)


APRE, %
144.5
130.1
115.6
101.2
86.70
72.25
57.80
43.35
28.90
14.45
0.000
Absolute Percent Relative Error, %
120
100
80
60
40
20
0
Temperature, K
MW n-Alkane

(e)


APRE, %
144.5
130.1
115.6
101.2
86.70
72.25
57.80
43.35
28.90
14.45
0.000
Absolute Percent Relative Error, %
120
100
80
60
40
20
0
Temperature, K
MW n-Alkane

(f)

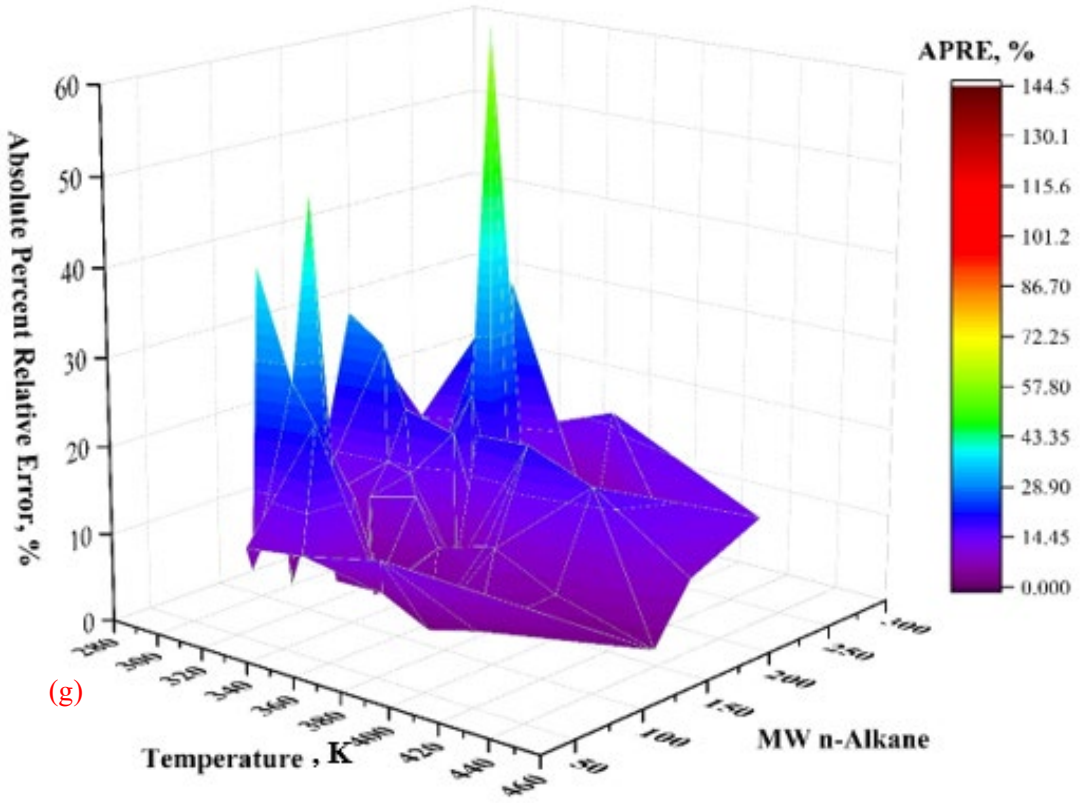


Figure S3. Absolute percent relative error (APRE) contour of IFT for: a) RBF-ICA, b) RBF-ACO, c) RBF-GA, d) RBF-PSO, e) RBF-GSA, f) MLP-LM, and g) GMDH at different temperatures and MWs of n-alkanes.

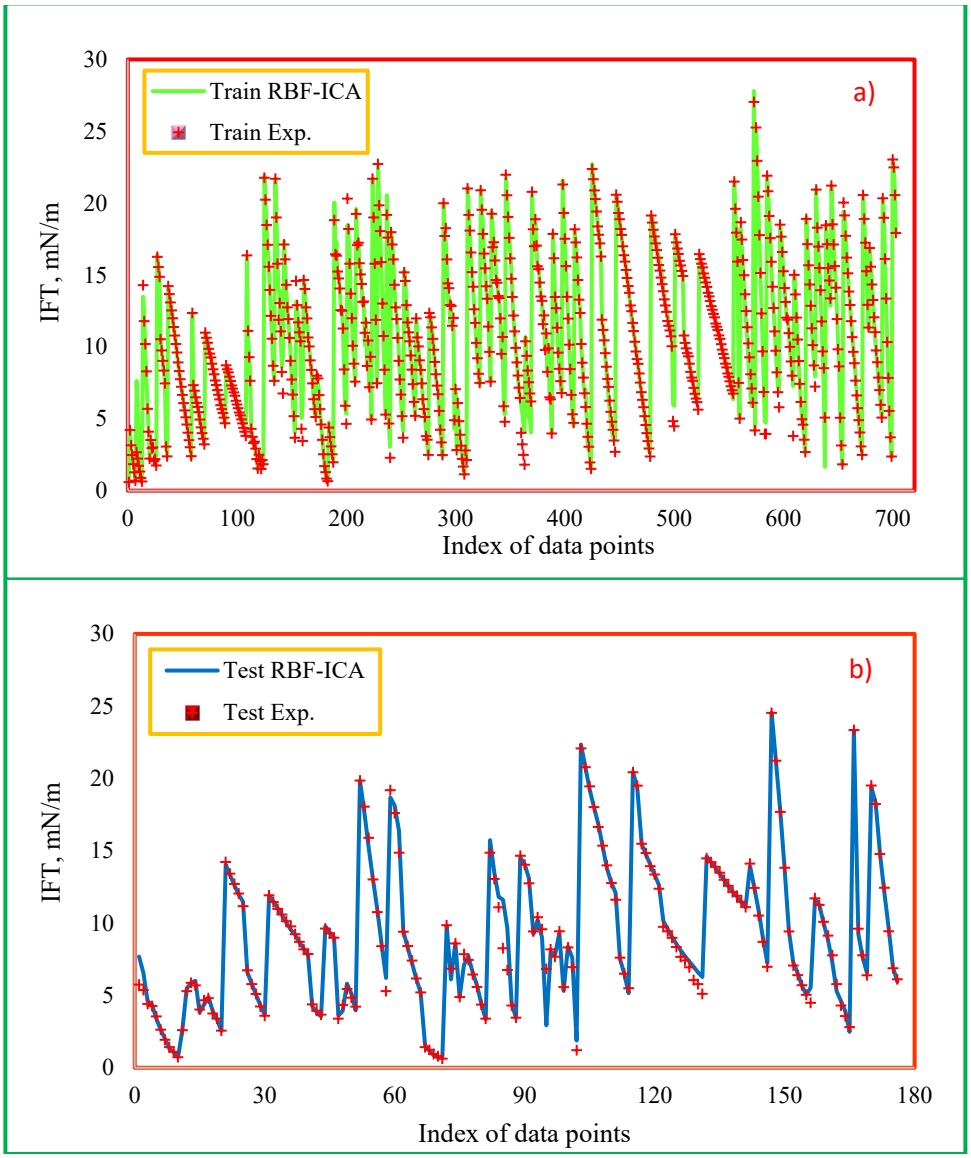


Fig S4. Experimental data versus RBF-ICA predictions of IFT between CO_2 and n-alkane: a) training subset, and b) testing subset

References

[1] Hagan MT, Menhaj MB. Training feedforward networks with the Marquardt algorithm. IEEE Trans Neural Networks 1994;5:989–93.

[2] Kişi Ö, Uncuoğlu E. Comparison of three back-propagation training algorithms for two case studies 2005.

[3] Bedekar PP, Bhide SR. Optimum coordination of directional overcurrent relays using the hybrid GA-NLP approach. IEEE Trans Power Deliv 2010;26:109–19.

[4] Alam MN, Das B, Pant V. A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination. Electr Power Syst Res 2015;128:39–52.

[5] Karkevandi-Talkhooncheh A, Hajirezaie S, Hemmati-Sarapardeh A, Husein MM, Karan K, Sharifi M. Application of adaptive neuro fuzzy interface system optimized with evolutionary algorithms for modeling CO2-crude oil minimum miscibility pressure. Fuel 2017;205:34–45.

[6] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci (Ny) 2009;179:2232–48.

[7] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. 2007 IEEE Congr. Evol. Comput., IEEE; 2007, p. 4661–7.

[8] Ansari HR. Use seismic colored inversion and power law committee machine based on imperial competitive algorithm for improving porosity prediction in a heterogeneous reservoir. J Appl Geophys 2014;108:61–8.

[9] Gholami A, Ansari HR, Hosseini S. Prediction of crude oil refractive index through optimized support vector regression: a competition between optimization techniques. J Pet Explor Prod Technol 2017;7:195–204.

[10] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci., Ieee; 1995, p. 39–43.

[11] Panigrahi BK, Shi Y, Lim M-H. Handbook of swarm intelligence: concepts, principles and applications. vol. 8. Springer Science & Business Media; 2011.

[12] Yazdanpanah A, Rezaei A, Mahdiyar H, Kalantarials A. Development of an efficient hybrid GA-PSO approach applicable for well placement optimization. Adv Geo-Energy Res 2019;3:365–74.

[13] Castillo O. Type-2 fuzzy logic in intelligent control applications. vol. 272. Springer; 2012.

[14] Onwunalu JE, Durlofsky LJ. Application of a particle swarm optimization algorithm for determining optimum well location and type. Comput Geosci 2010;14:183–98.

[15] Sharma A, Onwubolu G. Hybrid particle swarm optimization and GMDH system. Hybrid self-organizing Model. Syst., Springer; 2009, p. 193–231.

[16] Karkevandi-Talkhooncheh A, Rostami A, Hemmati-Sarapardeh A, Ahmadi M, Husein

MM, Dabir B. Modeling minimum miscibility pressure during pure and impure CO2 flooding using hybrid of radial basis function neural network and evolutionary techniques. Fuel 2018;220:270–82.

[17] Chen M-Y. A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering. Inf Sci (Ny) 2013;220:180–95.

[18] Lin C-J, Hong S-J. The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition. Neurocomputing 2007;71:297–310.

[19] Shi Y, Eberhart R. A modified particle swarm optimizer. 1998 IEEE Int. Conf. Evol. Comput. proceedings. IEEE world Congr. Comput. Intell. (Cat. No. 98TH8360), IEEE; 1998, p. 69–73.

[20] Chiou J-S, Tsai S-H, Liu M-T. A PSO-based adaptive fuzzy PID-controllers. Simul Model Pract Theory 2012;26:49–59.

[21] Stützle T, Hoos HH. MAX–MIN ant system. Futur Gener Comput Syst 2000;16:889–914.

[22] Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1997;1:53–66.

[23] Dorigo M. Optimization, learning and natural algorithms. PhD Thesis, Politec Di Milano 1992.

[24] Heris SMK, Khaloozadeh H. Ant colony estimator: an intelligent particle filter based on ACOR. Eng Appl Artif Intell 2014;28:78–85.

[25] Larrañaga P, Lozano JA. Estimation of distribution algorithms: A new tool for evolutionary computation. vol. 2. Springer Science & Business Media; 2001.

[26] Lozano JA, Larrañaga P, Inza I, Bengoetxea E. Towards a new evolutionary computation: advances on estimation of distribution algorithms. vol. 192. Springer; 2006.

[27] Socha K, Dorigo M. Ant colony optimization for continuous domains. Eur J Oper Res 2008;185:1155–73.