Applied Soft Computing 13 (2013) 3864–3872

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering

Cheng-Lung Huang*, Wen-Chen Huang, Hung-Yi Chang, Yi-Chun Yeh, Cheng-Yi Tsai

National Kaohsiung First University of Science and Technology, Information Management, 2, Juoyue Road, Nantz District, Kaohsiung 811, Taiwan

ARTICLE INFO

Article history:
Received 19 June 2011
Received in revised form 23 April 2013
Accepted 3 May 2013
Available online 31 May 2013

Keywords:
Swarm intelligence
Data clustering
Ant colony optimization
Particle swarm optimization
Hybrid systems

ABSTRACT

Ant colony optimization (ACO) and particle swarm optimization (PSO) are two popular algorithms in swarm intelligence. Recently, a continuous ACO named ACOR was developed to solve the continuous optimization problems. This study incorporated ACOR with PSO to improve the search ability, investigating four types of hybridization as follows: (1) sequence approach, (2) parallel approach, (3) sequence approach with an enlarged pheromone-particle table, and (4) global best exchange. These hybrid systems were applied to data clustering. The experimental results utilizing public UCI datasets show that the performances of the proposed hybrid systems are superior compared to those of the *K*-mean, standalone PSO, and standalone ACOR. Among the four strategies of hybridization, the sequence approach with the enlarged pheromone table is superior to the other approaches because the enlarged pheromone table diversifies the generation of new solutions of ACOR and PSO, which prevents traps into the local optimum.

1. Introduction

Swarm intelligence (SI) originated from the study of colonies or swarms of social organisms [1]. Among various algorithms in swarm intelligence, the ant colony algorithms (ACO) and particle swarm optimization (PSO) are two commonly used metaheuristic algorithms. In recent years, a large number of related studies have proposed methodologies and applications of the two algorithms.

The ant-based algorithm, chiefly used to solve combinatorial optimization problems, was inspired by observations of the foraging behavior of real ants [2,3]. ACO-related researches in various areas are numerous [4–7]. Traditional ACO mainly tackles combinatorial optimization problems. For applying traditional ACO in continuous-valued optimization problems, the continuous variables are usually discretized into discrete variables prior to performing ACO to handle the continuous-valued optimization problems [8,9]. This approach does not generally suffice in accuracy because the length of the discretization interval affects the quality of solutions. Recently, Socha and Dorigo [10] proposed a new ant-based algorithm named ACOR to solve continuous optimization problems, which has attracted research attention [11–13].

PSO is inspired by social behavior among individuals, for instance, bird flocks. Particles (individuals) representing a potential problem solution move through the search space [14]. Each particle produces a new acceleration to change the current position of the particle according to three parameters: (1) the best value of the particle itself; (2) the global best value; and (3) the acceleration of the particle. This approach, after several iterations, can find the optimal solution.

Premature convergence that leads to a fall into local optimum may exist in metaheuristic algorithms including ACOR and PSO. Premature convergence may be caused by a lack of diversity in the searching process. To improve diversity in the searching process, certain studies have proposed the concept of multiple sub-swarms with different characteristics [15,16], and others have proposed hybrid systems that combine different swarm algorithms [17]. Being applied as a part of a large system, the hybridization can improve the original algorithm and obtain a superior solution quality. Generally, the performance of a single algorithm is inferior to that of the hybrid algorithm [18,19]. Hybridization in swarm intelligence is essential for performance enhancement of a swarm intelligence optimization algorithm.

Hybridization of discrete ACO and PSO is popular; nevertheless, the hybridization of ACOR and PSO has not yet been investigated. This study proposes an innovative hybrid model by combining ACOR and PSO, and four types of hybridization strategies to combine the pheromone table and particle swarm. To demonstrate the computational performances, the proposed hybrid models are applied to data clustering, which is a difficult task with continuous-valued variables. Clustering divides large data objects into several clusters with a small similarity of intra-clusters and a large similarity of inter-clusters. Data clustering is applied to different

* Corresponding author. Tel.: +886 76011000.
E-mail address: clhuang@nkfust.edu.tw (C.-L. Huang).

1568-4946/$ – see front matter
http://dx.doi.org/10.1016/j.asoc.2013.05.003

business areas where large transactional and customer data are accumulated, including sales management, marketing analysis, and document management.

Swarm intelligence algorithms including ant clustering and particle swarm optimization has been applied to clustering in recent years. To date, numerous studies have applied data clustering by employing ACO or PSO [20,21]. The ACOR, which is designed for continuous numerical optimization problems, is applicable in data clustering; however, the application of ACOR, or ACOR combined with PSO in data clustering, is lacking. In the proposed methodology, this study investigates several approaches for the combination of the pheromone table and the solution of the particle swarm.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature including the basic ACO, ACOR and PSO algorithms; Section 3 describes the ACOR–PSO hybrid systems; Section 4 presents the experimental results from the UCI dataset; and lastly, Section 5 provides conclusive remarks.

2. Related research

2.1. Ant colony optimization

Ant colony optimization (ACO) is an artificial system inspired by the behavior of real ant colonies, and is applied to solve discrete combinatorial optimization problems. The first ACO was developed to solve the classical traveling salesman problem (TSP) [22,2,23]. In the standard ACO, ants make a probabilistic choice based on the transition probability before updating the pheromones along their trail to a food source. In the TSP problem, the transition probability from City i to City j for the kth ant at the time step t is expressed as follows:

$$PROB_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{j \in I_i^k} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}} & \text{if } j \in I_i^k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\tau_{ij}(t)$ is the amount of pheromone trail on edge (i, j) at the time t; η_{ij} is the a priori available heuristic information; α and β are two factors that specify the relative effects of the pheromone trail and heuristic information; and I_l^k is the set of feasible neighborhood cities that have not yet been visited by the ant k.

After each ant has completed a tour, the pheromone trails are updated by initially lowering them with a constant evaporation rate, and successively allowing each ant to deposit pheromone on the arcs that are a part of its tour, as indicated in the following equation:

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{M} \Delta\tau_{ij}^{k} \tag{2}$$

where M is the number of ants, and ρ is the pheromone trail evaporation rate ($0 < \rho < 1$). The parameter ρ is used to prevent unlimited accumulation of the pheromone trails and enables the algorithm to “forget” previously made bad decisions. On arcs that are not selected by the ants, the associated pheromone strength declines exponentially with the number of iterations. $\Delta\tau_{ij}^{k}$, the quantity per unit of length of the trail substance that is laid on the edge (i, j) by the kth ant, is defined as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if ant } k \text{ uses edge } (i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where L_k denotes the tour length, and Q is a predefined constant.

2.2. Ant algorithm for continuous domain

Socha and Dorigo [10] proposed the ant algorithm called ACOR for continuous domain. Following this research, certain studies focused on applying ant colony search to solve continuous optimization problems [24–28]. In the ACOR, each row in the pheromone table represents a solution of a set of decision variables. Each solution has a value of objective function. The new ants of next generation are generated using a roulette wheel probability based on the objective function of each solution in the pheromone table. The detail algorithm of ACOR is as follows:

Step 1: For each solution s_i in the pheromone table, calculate the value of the objective function $f(s_i)$. Sort the solutions in the pheromone table according to their objective values, that is, for a minimum problem: $f(s_1) \le f(s_2) \le \ldots \le f(s_i) \le \ldots \le f(s_K)$, where K is number of rows (solutions) in the pheromone table.
Step 2: Calculate the weight ω, so that $\omega_1 \ge \omega_2 \ge \ldots \ge \omega_i \ge \ldots \ge \omega_K$.

$$\omega_i = \frac{1}{qK\sqrt{2\pi}} e^{-(i-1)^2/2q^2K^2} \tag{4}$$

where q represents the learning rate between 0 and 1.
Step 3: Compute the probability of the roulette wheel p_i, according to the value of ω_i for each solution.

$$p_i = \frac{\omega_i}{\sum_{j=1}^{K} \omega_j} \tag{5}$$

Step 4: Repeat the following steps M times to generate M new ants ($M \le K$): produce a new value repeatedly for each variable of a new ant by employing the normal distribution $N(\mu_i^d, \sigma_i^d)$, where μ_i^d is a value selected from the dth value (variable) of the ith solution in the pheromone table by the probability of p_i, and σ_i^d is defined as follows:

$$\sigma_i^d = \xi \sum_{j=1}^{K} \frac{|x_j^d - x_i^d|}{K-1} \tag{6}$$

where x_i^d is the value of the dth variable of the ith solution, K is the size of the pheromone table, and ξ represents the evaporation rate.
Step 5: Evaluate the M new ants and replace the inferior solutions in the pheromone table by the superior solutions from the M new ants.

2.3. Particle swarm optimization

PSO is inspired by the social behavior among individuals. Recently, many PSO variants were introduced including [29–31]. Particles representing a potential solution to the optimization problem move through a search space. Each particle maintains a record of the position of its previous optimal performance called *pbest*. The optimal performance obtained thus far from the entire swarm is call *gbest*. A particle calculates its new velocity and updates its new position based on the direction of its previous optimal position and the global best position. Let $pbest_i^d$ denote the previous optimal position of the dth dimension encountered by the ith particle, and $gbest^d$ denote the global best position of the dth dimension. The current velocity of the dth dimension of the ith particle at the iteration t is defined as follows:

$$\begin{aligned} v_i^d(t) = {} & w \times v_i^d(t) + c_1 \times rnd \times (pbest_i^d - x_i^d(t-1)) \\ & + c_2 \times rnd \times (gbest^d - x_i^d(t-1)), \end{aligned} \tag{7}$$

$$v_i^d \in [-v_{\max}, v_{\max}] \tag{8}$$

In the abovementioned formula, *rnd* is a random function in the range [0,1]. The positive constants c_1 and c_2 are the personal and social learning factors, respectively, and w is the inertia weight, which was first introduced by Shi and Eberhart [32]. Inertia weight balances the global exploration and local exploitation. The velocity is restricted to the $[-v_{\max}, v_{\max}]$ range, in which $v_{\max}$ is a predefined boundary value. The value of $v_{\max}$ determines the resolution of the search regions between the present and target position. Eberhart and Shi [33] suggested that $v_{\max}$ should be set at approximately 10–20% of the dynamic range of the variable in each dimension. The new position of a particle is calculated with the following formula:

$$x_i^d(t) = x_i^d(t-1) + v_i^d(t) \tag{9}$$

2.4. Hybrid systems

Studies on hybrid systems in swarm intelligence have recently gained popularity, for instance, those on the integration of PSO with ACO [34–38], ACOR with Hooke and Jeeves local search [39], and ACOR with differentia evolution [39]. The hybrid application combining the ant colony algorithm and particle swarm algorithm is popular. To our knowledge, the hybrid system of mixing the continuous ant colony and particle swarm algorithm has not yet been implemented. This paper investigates the strategy for combining the continuous ant colony algorithm and the particle swarm optimization to improve data clustering performance.

2.5. Data clustering

Clustering is an unsupervised data segmentation technique for grouping a set of data objects into classes of similar data objects. Certain popular clustering methods can be adopted such as partitioning methods, hierarchical methods, grid-based methods, model-based methods, and density-based methods [40]. Swarm intelligence, including ant algorithms and PSO, has been used in clustering. Ant-based clustering was first introduced by Deneubourg et al. [41] to model ants gathering items to form heaps; other studies have proposed improved versions [42–46]. Another ant-based approach in clustering involves representing a solution as a set of cluster centers, and adopting the ant algorithm to search for the optimal set of cluster centers [38,47].

Several studies have applied PSO in clustering [48–53]. These literatures showed that with certain enhancements, PSO can avoid trapping into local optimum, and outperformed a few other traditional clustering algorithms.

3. Hybridization architecture of ACOR and PSO

3.1. Cluster center representation and data preprocessing

In our proposed ACOR–PSO hybridization, a solution is represented as a combination of cluster centers. Thus, the length of a solution is equal to the dimensions of a dataset multiplying the number of clusters. The objective function, which is used to evaluate the merit of clustering, is defined as

$$f = \frac{\sum_{k=1}^{N_c}\sum_{i=1}^{N_s} \min ||X_i - C_k||^2}{\sum_{k,j=1;k\neq j}^{N_c} d(C_k, C_j)} \tag{10}$$

where N_c = number of centers, N_s = sample size, $||X_i - C_k||^2$ = distance between sample i to center k, $d(C_k, C_j)$ = distance between center k and center j.

The intra-cluster distance is obtained by summarizing all pairwise distances from points in the cluster to the cluster center. The inter-cluster distance between clusters is computed as the distance between the centers. Clusters must be compact, with cluster centers far from each other for good clustering, giving f a small value.

All the variables are scaled before clustering. The main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Each variable can generally be linearly scaled to the range [0,1], as with the following formula:

$$x^{new} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{11}$$

where x^{new} is the scaled value of the variable x, and $\min(x)$ and $\max(x)$ are the minimum and maximum values of the variable x, respectively.

3.2. Four strategies of combination for ACOR and PSO

This study hybridizes ACOR and PSO in clustering. In the ACOR, the new ants are generated based on the pheromone table; in the PSO, the new position for each particle is produced based on the current position of the particle. The ACOR (*ACOR-Module*) provides new solutions at each ant cycle (iteration), where new ants are generated based on the current solutions of the pheromone table, abbreviated as "*PHERO*". A pseudocode of the *ACOR-Module* is presented as follows:

Algorithm: *ACOR-Module*
Do for each ant, ant_m, m = 1,2, . . ., M
  Do for each variable x_m^d in ant_m, d = 1,2, . . ., D
    (1) Choose a μ_i^d from the pheromone table, *PHERO* (K dimensions), based on the probability p_i in Eq. (5), where $i \in \{1,2, \ldots, K\}$
    (2) Calculate σ_i^d as follows: if $rnd \le \varnothing_1$, generate a standard deviation σ_i^d by Eq. (6). Else, generate a random standard deviation σ_i^d using the uniform distribution $U(0,1)$, where rnd is a random number ($0 \le rnd \le 1$), and $\varnothing_1$ is a predefined threshold between 0 and 1 to generate a random value for σ_i^d
    (3) Produce a new value of x_m^d as follows: if $rnd \le \varnothing_2$, generate a new value by the normal distribution $N(\mu_i^d, \sigma_i^d)$. Else, generate a random value by the uniform distribution $U(0,1)$, where rnd is a random number ($0 \le rnd \le 1$), and $\varnothing_2$ is a predefined threshold between 0 and 1 for generating a random solution for x_m^d
  EndDo
EndDo
Return new ants with M rows (solutions)

In the *ACOR-Module*, this study introduces two thresholds to improve the exploration ability: $\varnothing_1$ for generating a random value of standard deviation σ_i^d, and $\varnothing_2$ to generate a random value of the dth variable of the mth ant. A proper setting of the two thresholds via experiments is required.

At the iteration t, PSO (*PSO-Module*) generates new particles based on the current positions of particles, abbreviated as *PAR*. The basic procedure for the *PSO-Module* is as follows:

Algorithm: *PSO-Module*
Do for each $particle_i$ in the *PAR*, i = 1,2, . . ., K
  Do for each dimension d in $particle_i$, d = 1,2, . . ., D
    Calculate the new velocity $v_i^d(t)$ by Eq. (7)
    Update the new position $x_i^d(t)$ by Eq. (9)
  EndDo
EndDo
Return new positions of particles, *PAR*

The *ACOR-Module* and *PSO-Module* are thus combined via four hybridization strategies according to the update strategy for the pheromone table and current position of the particle swarm. The four strategies of hybridization are proposed as follows: (1) sequence approach; (2) parallel approach; (3) sequence with double the size of the pheromone table; and (4) global best exchange. The first three strategies combine the pheromone table and PSO

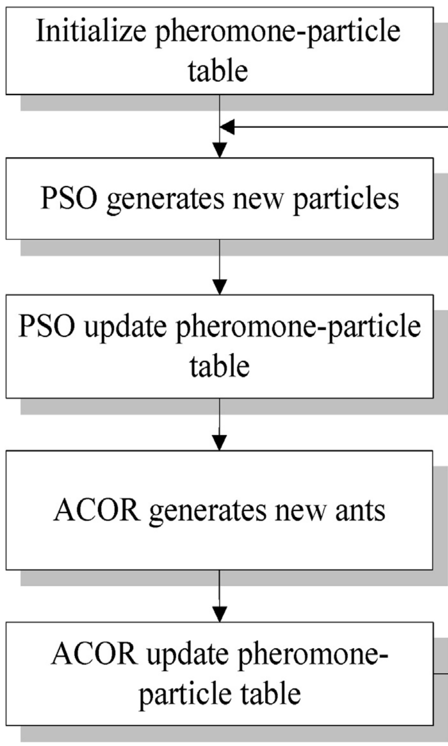

Fig. 1. Sequence approach.

swarm in different manners, while the fourth strategy simply exchanges the global best solution of each algorithm.

3.2.1. Sequential approach

In the sequential approach, the ACOR and PSO share the same set of solutions, named the "pheromone-particle" table, abbreviated as "*PHERO_PAR*", which acts as the pheromone in ACOR and the current solution information in PSO. Based on the pheromone-particle table, the PSO generates new particles and replaces the inferior solutions in the pheromone-particle table with superior solutions from the new particles. Based on the updated pheromone-particle table by the PSO, the ACOR subsequently generates new ants and replaces the inferior solutions in the pheromone-particle table with superior solutions from the new ants. The features of sequential approach are: (1) the superior solutions generated by the PSO and the ACOR can be retained in the pheromone table; and (2) PSO generates new ants and replaces the inferior solutions in the pheromone table; this may diversify the pheromone table, thus preventing the ACOR from trapping into the local optimum. Fig. 1 shows the main steps of the sequential update approach, and the detailed algorithm is as follows:

Algorithm: *Sequential approach*
Create and initialize a pheromone-particle table, *PHERO_PAR*, with K rows (particles) and D dimensions using uniform distribution
Calculate fitness using Eq. (10) for each particles in *PHERO_PAR*, update *pbest* for each particle, and update *gbest* with the best fitness value of all the particles
Do until the stopping condition is true
  Generate new positions for all particles in *PHERO_PAR* by performing *PSO-Module*
  Do for each $particle_i$ in *PHERO_PAR*, $i = 1,2, \ldots, K$
    Calculate fitness of $particle_i$ using Eq. (10), and update its *pbest* if the fitness value of $particle_i$ is better than its *pbest*
    Update *gbest* if the fitness value of $particle_i$ is better than the *gbest*
  EndDo
  Generate new ants, *ANT*, with M rows by performing *ACOR-Module* based on the updated *PHERO_PAR* performed in the previous step
  Do for each ant_i in the *ANT*, $i = 1,2, \ldots, M$
    Calculate fitness of ant_i using Eq. (10)
    Replace the worst solution in *PHERO_PAR* and update its *pbest*, if the fitness value of ant_i is better than that of the worst particle in *PHERO_PAR*
    Update *gbest*, if the fitness value of ant_i is better than the *gbest*
  EndDo
Enddo
Return *gbest*, the optimal solution

3.2.2. Parallel approach

Based on the pheromone-particle table, the PSO generates new particles, and the ACOR generates new ants in parallel. The inferior solutions in the pheromone-particle are replaced by the superior solutions from both the K new particles and M new ants. The main difference between the sequential and parallel approach is the manner in which they update the pheromone-particle table. In the parallel approach, the new ants are generated directly from the pheromone-particle table, without being based on the new particles, whereas in the sequential approach, the ants are generated from the updated pheromone-particle that is renewed by the PSO. The diversity of the pheromone table may be different between the two approaches, and this may result in different new solutions. Fig. 2 shows the main concept of the parallel update approach, and its detailed algorithm is as follows:

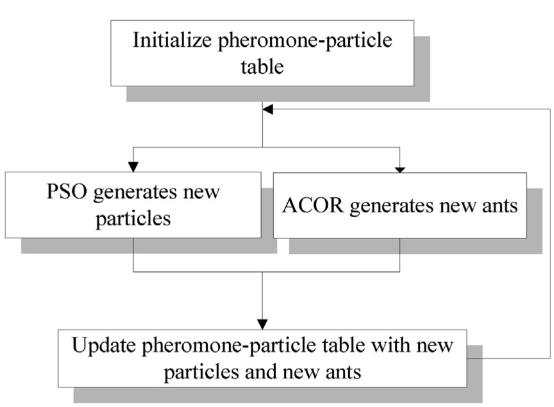

Fig. 2. Parallel approach.

Algorithm: *Parallel approach*
Create and initialize a pheromone-particle table, *PHERO_PAR*, with K rows and D dimensions using uniform distribution
Calculate fitness using Eq. (10), update *pbest* for each particle, and update *gbest* with the best fitness value of all the particles
Do until the stopping condition is true
  Generate new ants, *ANT*, with M rows by performing *ACOR-Module* based on *PHERO_PAR*
  Generate new positions for particles in *PHERO_PAR* by performing *PSO-Module*
  Do for each $particle_i$ in *PHERO_PAR*, $i = 1,2, \ldots, K$
    Calculate fitness of $particle_i$ using Eq. (10), and update its *pbest* if the fitness value of $particle_i$ is better than its *pbest*
    Update *gbest* if the fitness value of $particle_i$ is better than the *gbest*
  EndDo
  Do for each ant_i in *ANT*, $i = 1,2, \ldots, M$
    Calculate fitness of ant_i using Eq. (10)
    Replace the worst solution in *PHERO_PAR* and update its *pbest*, if the fitness value of ant_i is better than that of the worst solution in *PHERO_PAR*
    Update *gbest*, if the fitness value of ant_i is better than the *gbest*
  EndDo
Enddo
Return *gbest*, the optimal solution

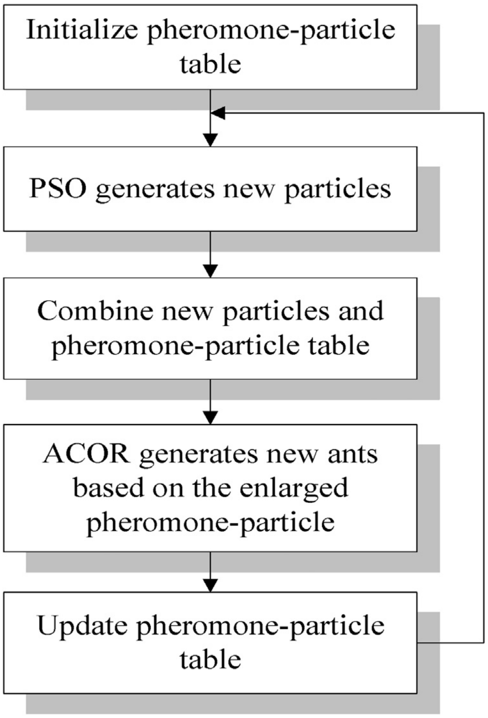

Fig. 3. Sequence approach with enlarged pheromone-particle table.

3.2.3. *Sequential approach with the enlarged pheromone-particle table*

Based on the pheromone-particle table, the PSO generates *K* new particles, which are successively combined with the pheromone-particle table to form an enlarged pheromone-particle table with the size of 2*K*. Based on the enlarged pheromone-particle table, the ACOR generates *M* new ants. Among the new ants and the enlarged pheromone-particle table, the best *K* solutions are kept to form a new pheromone-particle table. For the PSO, the *pbest* and *gbest* are obtained from the updated pheromone-particle table; for the ACOR, the new ants are generated based on the enlarged pheromone table with diversity; thus, this approach prevents trapping into a local optimum. Fig. 3 shows the main concept of the sequential approach with the enlarged pheromone-particle table, and its detailed algorithm is as follows:

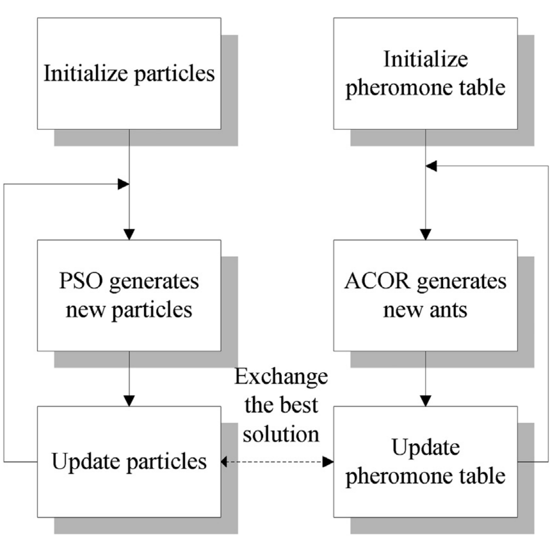

Fig. 4. Global best exchange approach.

Algorithm: *Sequential approach with enlarged pheromone-particle table*
Create and initialize a pheromone-particle table, *PHERO_PAR*, with *K* rows and *D* dimensions using uniform distribution
Calculate fitness using Eq. (10), update *pbest* for each particle, and update *gbest* with the best fitness value of all the particles
Do until the stopping condition is true
  Make a copy of *PHERO_PAR′* from *PHERO_PAR*, and generate new positions of particles in *PHERO_PAR′* by performing *PSO-Module*
  Do for each $particle_i$ in *PHERO_PAR′*, $i = 1,2, \ldots, K$
    Calculate fitness of $particle_i$ using Eq. (10), and update its *pbest* if the fitness value of $particle_i$ is better than its *pbest*
    Update *gbest* if the fitness value of $particle_i$ is better than the *gbest*
  EndDo
  Combine *PHERO_PAR′* and *PHERO_PAR* to form an enlarged pheromone table with $2 \times K$ rows, named as *PHERO_PAR″*
  Generate new ants, *ANT*, with *M* rows by performing *ACOR-Module* based on the *PHERO_PAR″*
  Do for each ant_i in the *ANT*, $i = 1,2, \ldots, M$
    Calculate fitness of ant_i using Eq. (10)
    Replace the worst solution in *PHERO_PAR′* and update its *pbest*, if the fitness value of ant_i is better than that of the worst solution in *PHERO_PAR′*
    Update *gbest*, if the fitness value of ant_i is better than the *gbest*
  EndDo
  Replace *PHERO_PAR* by *PHERO_PAR′*
Enddo
Return *gbest*, the optimal solution

3.2.4. *Global best exchange*

This strategy simply shares only the best solution instead of the entire pheromone/particle table with each other, as shown in Fig. 4. The PSO generates new particles based on its own particle table, and the ACOR generates new ants based on its own pheromone table. The two models exchange their best solution. This approach maintains the original features of the PSO and ACOR respectively. The algorithm of the global best exchange approach is detailed as follows:

Algorithm: *Global best exchange*
Create and initialize a particle table, *PAR*, with *K* rows and *D* dimensions using uniform distribution
Calculate fitness for each particle in *PAR* using Eq. (10), and update *pbest* for each particle
Update *gbest* with the best fitness value in *PAR*
Create and initialize a pheromone table, *PHERO*, with *K* rows and *D* dimensions using uniform distribution
Calculate fitness for each ant in *PHERO* using Eq. (10)
Update *bestOfAnt* with the best fitness value in *PHERO*. (Let the global best fitness of the ants is represented as *bestOfAnt*)
Update *globalOpt* with the best value of *gbest* and *bestOfAnt*. (Let the global optimal solution of ACOR and PSO is represented as *globalOpt*)
Do until the stopping condition is true
  Generate new positions of *PAR* by performing *PSO-Module*
  Do for each $particle_i$ in *PAR*, $i = 1,2, \ldots, K$
    Calculate fitness of $particle_i$ using Eq. (10), and update its *pbest* if the fitness value of $particle_i$ is better than the *pbest* in the previous iteration
    Update *gbest* if the fitness value of $particle_i$ is better than the *gbest*
  EndDo
  Generate new ants, *ANT*, with *M* rows by performing *ACOR-Module* based on the *PHERO*
  Do for each ant_i in the *ANT*, $i = 1,2, \ldots, M$
    Calculate fitness of ant_i using Eq. (10)
    Replace the worst solution in *PHERO*, if the fitness value of ant_i is better than that of the worst solution in *PHERO*
    Update *bestOfAnt*, if the fitness value of ant_i is better than the *bestOfAnt*
  EndDo
  Exchange global best as follows: replace the worst particle in the *PAR* with the *bestOfAnt*, and replace the worst ant in *PHERO* with the *gbest*
  Update *globalOpt* with the best value from the *gbest* and *bestOfAnt*
Enddo
Return *globalOpt*, the optimal solution

Table 1
Datasets from the UCI repository.

No.	Names	#Instances	Numeric features
1	German (credit card)	1000	24
2	Pima-Indian diabetes	768	8
3	Breast cancer (Wisconsin)	699	10
4	Iris	150	4
5	Yeast	1484	9
6	Wine	178	13
7	Ecoli	336	8
8	Liver disorders	354	6
9	Contraceptive method choice	1473	9

3.2.5. Computational complexity analysis

The time complexity of the aforementioned four hybridizations of the ACOR–PSO models are proportional to the number of iterations, the number of particles, and the number of ants, and can be computed according to their main steps, as follows:

$$N_{Iterations} \times (N_{NewAnts} \times T_{Ant} + N_{NewParticles} \times T_{Particle} + T_{Update}) \quad (12)$$

where $N_{Iterations}$: number of iterations, $N_{NewAnts}$: number of new ants, $N_{NewParticles}$: number of new particles, T_{Ant}: runtime for generating a new ant, $T_{Particle}$: runtime for generating a new particle, T_{Update}: runtime for relevant processes including updating the pheromone table, updating the position of a new particle, and other necessary processing times.

4. Experiments

4.1. Description of experiments

This study conducted experiments to evaluate the clustering performance of the proposed hybrid models on nine real-world datasets from the UCI database (see Table 1). This study evaluated four types of hybrid approaches, and compared the hybrid approaches between *K*-means, the PSO, and the ACOR.

The experiment was conducted as follows: perform clustering with a cluster number from 2 to 10; each clustering experiment is repeated five times. Thus, $9 \times 5 = 45$ experiments are conducted for each dataset. The clustering performances are calculated by using Eq. (10). The performances of the five repeated runs are averaged before the averaged performances for the 45 experiments are further averaged for each dataset.

Because the number of iterations affects clustering performances, each experiment is set to the same number of evaluations for a fair comparison on performance. A new solution is evaluated using the objective function, Eq. (10). An "evaluation" is defined as calculating the objective function for a solution. Therefore, as shown in Table 2, the number of evaluations for the PSO is equal to the size of particle swarm multiplied by the number of iterations, the number of ants multiplied by the number of iterations for the ACOR, and the size of the pheromone-particle table, in addition to the number of ants multiplied by the number of iterations for the hybrid models. Therefore, for the hybrid models, the number of evaluations is $(15 + 5) \times 500 = 10{,}000$, which is the same as that of the standalone ACOR or PSO. Among the four hybrid models, their number of evaluations is also the same; for the sequential approach with the enlarged pheromone-particle table, the new ants are generated from the enlarged pheromone-particle table without increasing its number of evaluations.

Table 2
Calculation of the number of evaluations.

Models	Evaluations
PSO	Size of swarm particles × number of iterations
ACOR	Number of ants × number of iterations
Hybrid models	(Size of pheromone-particle table + number of ants) × number of iterations

Table 3
System parameter setting.

Parameter	PSO	ACOR	Hybrid models
Learning rate C1	1.5		1.5
Learning rate C2	1.5		1.5
Learning rate q		0.98	0.98
Evaporation rate		0.05	0.05
Pheromone table		60	
Number of particles	20		
Size of pheromone-particle table			15
Number of ants		20	5
Number of iterations	500	500	500
Number of evaluations	10,000	10,000	10,000

The number of evaluations was set to 10,000 in this study. Other relevant system parameters for each model were fine-tuned by preliminarily experiments, and are shown in Table 3.

4.2. Comparisons against traditional models

This paper performed experiments in data clustering for the hybrid models, traditional *K*-means, the PSO, and the ACOR. For the sake of brevity, the hybridization of the sequence approach is abbreviated as Hybrid I, that of the parallel approach as Hybrid II, the sequence with the enlarged size of the pheromone table as Hybrid III, and the global best exchange as Hybrid IV. Table 4 illustrates the average 5-run performances with a cluster number from 2 to 10 by using the Iris dataset for *K*-means, the PSO, the ACOR and the four types of the hybrid model. The 5-run performances of the various number of clusters are further averaged for the nine experimental datasets, which are shown in Table 5.

Among *K*-means, the PSO, and the ACOR, the ACOR has a superior average performance than the PSO in all the datasets except the breast cancer and the wine recognition datasets. The *K*-mean clustering is inferior to the PSO and ACOR models, especially in datasets with large sample sizes including Liver, Pima, Yearst, Contraceptive, and German datasets.

For the proposed hybridization strategies, the performances of the four ACOR–PSO hybrid models are superior to those of the traditional *K*-means, the standalone PSO, and the standalone ACOR in all the datasets, except for the wine dataset – the average performance of the parallel update model (Hybrid II) is slightly inferior to that of the PSO in the wine dataset. However, experimental results using public UCI datasets generally show that the performances of the proposed hybrid systems are superior to those of the *K*-mean, the PSO, and the ACOR.

With experimental data from 2 to 10 clusters, this study conducted performance comparisons by employing the nonparametric Wilcoxon test between the four ACOR–PSO hybrid models and the other three models (*K*-means, the PSO, and the ACOR). The reasons for adopting the nonparametric test with dependent samples to compare the performances are as follows: if population distributions are unknown and samples are identical or dependent, the nonparametric test with dependent samples is suggested for model comparison [54,55]. Table 6 shows the results of 2-tailed significance levels using the Wilcoxon test. As shown in the table, because most comparisons achieve a significance level of 0.1 except the values marked in gray, in most cases, the hybrid models notably outperform the other three models with a 0.1 significance level.

4.3. Comparisons between various hybrid models

The performance comparisons using the nonparametric Wilcoxon signed ranks test between the four ACOR–PSO hybrid

Table 4
Average 5-run performances using Iris dataset for *K*-means, PSO, ACOR and four types of hybrid models.

#Cluster	*K*-means	PSO	ACOR	Hybrid I	Hybrid II	Hybrid III	Hybrid IV
2	13.0206	12.0466	11.9031	11.8959	11.8956	11.8956	11.9010
3	2.9166	2.8395	2.7470	2.7191	2.7203	2.7179	2.7248
4	1.3438	1.3978	1.3331	1.1912	1.2373	1.1937	1.2433
5	0.7036	0.6981	0.7432	0.6238	0.6285	0.6148	0.6491
6	0.4373	0.4578	0.4709	0.3813	0.4105	0.3537	0.3913
7	0.2781	0.2979	0.3352	0.2722	0.2878	0.2429	0.2692
8	0.1947	0.2164	0.2562	0.2030	0.2173	0.1706	0.1923
9	0.1633	0.1558	0.1974	0.1399	0.1603	0.1209	0.1427
10	0.1049	0.1146	0.1668	0.1051	0.1256	0.0941	0.1070
Average	2.1292	2.0249	2.0170	1.9480	1.9648	1.9338	1.9579

Table 5
Summaries of average performances for *K*-means, PSO, ACOR and four types of hybrid models.

Dataset	*K*-means	PSO	ACOR	Hybrid I	Hybrid II	Hybrid III	Hybrid IV
Breast cancer	30.9345	28.9260	29.2482	26.9591	27.1244	26.0862	26.9050
Ecoli	14.8660	9.4496	7.9861	7.6331	7.8261	7.3992	8.0180
Liver	11.4383	4.7926	4.6849	4.4694	4.5044	4.3624	4.5133
Pima	45.6638	17.5309	17.4704	16.4040	16.9912	16.0425	16.4317
Yearst	67.7618	20.8300	18.0631	16.7473	17.4590	16.2527	17.4873
Contraceptive	175.3589	120.1819	113.2069	107.6461	110.2654	105.7992	109.3104
Iris	2.1292	2.0249	2.0170	1.9480	1.9648	1.9338	1.9579
Wine	14.5580	10.1253	10.1806	9.6229	10.6740	9.1561	9.6060
German	355.7430	207.6890	196.2553	181.9553	192.6266	174.1519	180.2570

Table 6
Performance comparison among models using the Wilcoxon test.

Dataset	Hybrid models vs. *K*-means				Hybrid models vs. PSO				Hybrid models vs. ACOR			
	I	II	III	IV	I	II	III	IV	I	II	III	IV
Breast cancer	0.086	0.594	0.028	0.028	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008
Ecoli	0.021	0.010	0.008	0.028	0.008	0.008	0.008	0.008	0.008	0.110	0.008	0.953
Liver	0.008	0.008	0.008	0.008	0.008	0.110	0.008	0.139	0.008	0.008	0.008	0.008
Pima	0.008	0.008	0.008	0.008	0.008	0.314	0.011	0.139	0.008	0.015	0.008	0.008
Yearst	0.008	0.008	0.008	0.008	0.008	0.021	0.008	0.008	0.008	0.021	0.008	0.008
Contraceptive	0.859	0.859	0.066	0.859	0.008	0.051	0.008	0.008	0.008	0.515	0.008	0.011
Iris	0.028	0.110	0.008	0.011	0.008	0.066	0.008	0.008	0.008	0.008	0.008	0.008
Wine	0.051	0.953	0.011	0.011	0.008	0.086	0.008	0.008	0.015	0.953	0.008	0.008
German	0.021	0.214	0.021	0.011	0.011	0.515	0.008	0.008	0.008	0.110	0.008	0.008

models are shown in Table 7. The significance between Hybrid III and the other three hybrid models achieves 0.05 for all the datasets except for the Iris (Hybrid IV) and Wine datasets (Hybrid I and IV). Generally, the performance of the sequential update model with the enlarged size of the pheromone table (Hybrid-III) is significantly superior to that of the other three hybrid models.

One possible reason for superior performance of the Hybrid-III is that the double size of the pheromone-particle table improves the diversity for generating new solutions. The diversity of new populations can improve the solution quality [17]. Some possible strategies may improve the diversity of populations including sub-populations [15,56], communication and migration between sub-populations [57,58], and random velocity in PSO [59]. The diversity of pheromone-particle table is another possible way investigated in this study to improve the diversity of populations.

Table 7
Performance comparison between four types of the hybrid model obtained via the Wilcoxon signed ranks test.

Dataset	Hybrid III vs. Hybrid I	Hybrid III vs. Hybrid II	Hybrid III vs. Hybrid IV
Breast cancer	0.008	0.011	0.008
Ecoli	0.008	0.008	0.008
Liver	0.008	0.008	0.008
Pima	0.038	0.021	0.028
Yearst	0.011	0.008	0.008
Contraceptive	0.028	0.011	0.015
Iris	0.066	0.008	0.678
Wine	0.374	0.008	0.260
German	0.021	0.012	0.008

The proposed implementation was executed on the Matlab 7 platform, with an Intel Core 2 Duo CPU running at 3.0 GHz and 4 GB RAM. The average CPU times for the four models required a couple of minutes. There were no significant differences between the four hybrid models.

4.4. Description of convergence

This section discusses the convergence of the proposed algorithm. The Iris dataset with ten cluster centers is regarded as an example. Fig. 5 shows the convergence chat for the PSO, the ACO, and the four ACOR–PSO hybrid models, where the *X*-axis represents the number of iterations, and the *Y*-axis is the average performance. In this example, the ACOR is more likely to trap into local optimum than the other models. Generally, the PSO-ACOR hybrid models are superior compared to the PSO and ACOR models. For the four hybrid PSO-ACOR models, Hybrid-III performs well because after 480 generations, its performance is superior to the other models.

The length of a solution is proportional to the number of clusters. A large number of clusters complicates the search space, and more execution iteration is required to obtain a satisfying solution. The experimental results in this study are based on the performances at iteration 500 for the ACOR, the PSO, or the ACOR–PSO hybridized

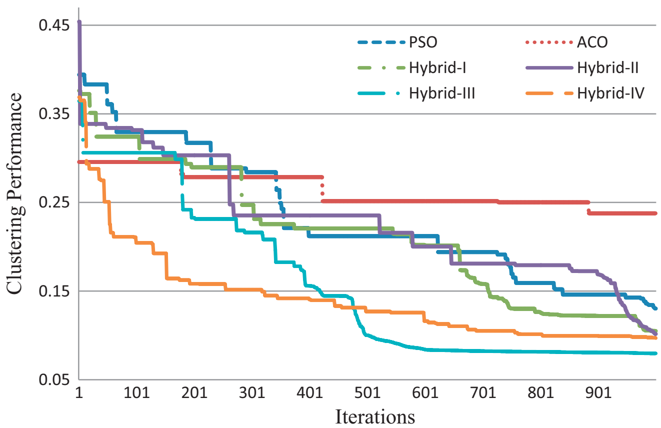

Fig. 5. Convergence illustrations for the PSO, the ACO, and the four ACOR–PSO hybrid models.

models. However, the clustering performance can be improved by increasing iterations. For example, for the Iris dataset, the performance using 10 clusters is 0.0941 at iteration 500 for the Hybrid III model. When the iteration increases to 1000, its performance is 0.0797, which is an improvement of approximately 15.3%.

5. Conclusion and discussions

Swarm intelligence focuses on the collective behaviors that result from the local interactions of the individuals and interactions with their environment. This study investigated the hybridizations of two popular swarm intelligence algorithms – the continuous ant colony algorithm and the particle swarm algorithm. Based on the strategy of hybridizing the pheromone table and particle swarm, updating the pheromone and particle, and exchanging the global best solutions, this study proposed four hybrid models: sequential, parallel, sequential with enlarged pheromone-particle table, and global best exchange.

Nine experimental datasets were tested for clustering. According to the average clustering performance, the hybrid models are superior to the traditional *K*-means, the standalone PSO, and the standalone ACOR models. The hybrid models can preserve diversity when generating new solutions.

The experimental results prove that the hybridization strategies are effective. Among the four hybrid models, the sequential update with the enlarged pheromone-particle table achieved the best average performance. In this hybridization strategy, the new particle solutions generated by the PSO are combined with the pheromone-particles table; thus increasing the chances of the ACOR to explore the search space. Compared to the standalone ACOR with a single pheromones table, the hybrid model has more diversity when generating new solutions, and might be capable of avoiding a fall into the local optimum.

This study summarizes the contributions of this research as follows: (1) this study is the first investigation on the hybridization strategies for the ACOR–PSO applied in data clustering. (2) The experiment indicates that, for designing the pheromone-particle table in the hybridized ACOR–PSO, the diversity was found to be critical for generating new solutions. A hybrid strategy that preserves diversity in the pheromone-particle table might lead to obtaining superior solutions after a certain number of iterations.

After the experimental assessment on the proposed ACOR–PSO hybridization strategies using various datasets, this study concludes and summarizes the contributions of this research as follows:

(1) The continuous ant algorithm ACOR is a new method that has recently been proposed. To our knowledge, the hybridization of ACOR with PSO has not yet been investigated. This study examined the hybridization strategies for the ACOR–PSO applied in data clustering, and provided promising experimental results.
(2) The proposed hybrid models have been proven to be superior compared to a standalone system in this study. Additionally, among the four strategies, the diversity of the pheromone-particle table was found to be critical for generating new solutions for the ACOR–PSO. A hybrid strategy that preserves diversity in the pheromone-particle table might lead to obtaining superior solutions after a certain number of iterations.

This study applied the proposed hybridization strategies of the ACOR–PSO in data clustering. Furthermore, they can be applied in other continuous-valued optimization problems, including function optimization. In addition to the extension of the application domain, future studies can investigate the model itself to improve its solution quality, such as by incorporating with other hybridization modes and adapting sub-swarm with different characteristics to enhance searching ability.

Acknowledgement

The authors would also like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 101-2221-E-327-026.

References

[1] A.P. Engelbrecht, Computational Intelligence: An Introduction, 2nd ed., John Wiley & Sons Ltd., New York, 2007.
[2] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.
[3] E. Bonabeau, M. Dorigo, G. Theraulaz, Inspiration for optimization from social insect behavior, Nature 406 (6) (2000) 39–42.
[4] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence from Natural to Artificial Systems, Oxford University Press, New York, 1999.
[5] R. Mullen, D. Monekoss, S. Barman, P. Remagnino, A review of ant algorithms, Expert Systems with Applications 36 (6) (2009) 9608–9617.
[6] Q. Duan, T.W. Liao, Improved ant colony optimization algorithms for determining project critical paths, Automation in Construction 19 (6) (2010) 676–693.
[7] T.W. Liao, Improving the accuracy of computer-aided radiographic weld inspection by feature selection, NDT&E International 42 (4) (2009) 229–239.
[8] C.L. Huang, ACO-based hybrid classification system with feature subset selection and model parameters optimization, Neurocomputing 73 (1–3) (2009) 438–448.
[9] X.L. Zhang, X.F. Chen, Z.J. He, An ACO-based algorithm for parameter optimization of support vector machines, Expert Systems with Applications 37 (9) (2010) 6618–6628.
[10] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185 (3) (2008) 1155–1173.
[11] S. Madadgar, A. Afshar, An improved continuous ant algorithm for optimization of water resources problems, Water Resources Management 23 (10) (2009) 2119–2139.
[12] A. Karimi, H. Nobahari, P. Siarry, Continuous ant colony system and tabu search algorithms hybridized for global minimization of continuous multi-minima functions, Computational Optimization and Applications 45 (3) (2010) 639–661.
[13] H. Liu, Z.X. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Applied Soft Computing 10 (2) (2010) 629–640.
[14] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.
[15] W.M. Spears, Simple subpopulation schemes, in: Proceedings of the Evolutionary Programming Conference, World Scientific, Vol. 3, Singapore, 1994, pp. 296–307.
[16] M. Lovbjerg, T.K. Rasmussen, T. Krink, Hybrid particle swarm optimizer with breeding and subpopulations, in: Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, USA, 2001, pp. 469–547.
[17] M. Lozano, C. García-Martínez, Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report, Computers & Operations Research 37 (3) (2010) 481–497.
[18] C. Blum, Ant colony optimization: Introduction and recent trends, Physics of Life Reviews 2 (4) (2005) 353–373.

[19] T.W. Liao, Two hybrid differential evolution algorithms for engineering design optimization, Applied Soft Computing 10 (4) (2010) 1188–1199.
[20] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann, New York, 2001.
[21] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, Cambridge, Massachusetts, 2004.
[22] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Transactions on System, Man, and Cybernetics 26 (1) (1996) 1–13.
[23] M. Dorigo, G.D. Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artificial Life 5 (2) (1999) 137–172.
[24] X.M. Hu, J. Zhang, Y. Li, Orthogonal methods based ant colony search for solving continuous optimization problems, Journal of Computer Science and Technology 23 (1) (2008) 2–18.
[25] A. Fetanat, G. Shafipour, Generation maintenance scheduling in power systems using ant colony optimization for continuous domains based 0–1 integer programming, Expert Systems with Applications 38 (8) (2011) 9729–9735.
[26] C.F. Juang, P.H. Chang, Designing fuzzy-rule-based systems using continuous ant-colony optimization, IEEE Transactions on Fuzzy Systems 18 (1) (2010) 138–149.
[27] C.F. Juang, P.H. Chang, Recurrent fuzzy system design using elite-guided continuous ant colony optimization, Applied Soft Computing 11 (2) (2011) 2687–2697.
[28] B. Ataie-Ashtiani, H. Ketabchi, Elitist continuous ant colony optimization algorithm for optimal management of coastal aquifers, Water Resources Management 25 (1) (2011) 165–190.
[29] R. Mendes, J. Kennedy, J. Neves, The informed particle swarm: simpler, maybe better, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 204–210.
[30] F. van den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176 (8) (2006) 937–971.
[31] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Computers and Operations Research 33 (3) (2006) 859–871.
[32] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, NJ, USA, 1998, pp. 69–73.
[33] R.C. Eberhart, Y. Shi, Particle swarm optimization: developments, application and resources, in: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea, 2001, pp. 81–86.
[34] N. Holden, A.A. Freitas, Hierarchical classification of G-protein-coupled receptors with a PSO/ACO algorithm, in: Proceeding of the IEEE Swarm Intelligence Symposium, Indiana, USA, 2006, pp. 77–84.
[35] Y. Meng, O. Kazeem, J.C. Muller, A hybrid ACO/PSO control algorithm for distributed swarm robots, in: Proceedings of the IEEE Swarm Intelligence Symposium, Hawaii, USA, 2007, pp. 273–280.
[36] X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu, L.M. Wang, An improved GA and a novel PSO-GA-based hybrid algorithm, Information Processing Letters 93 (5) (2005) 255–261.
[37] T. Niknam, An efficient hybrid evolutionary algorithm based on PSO and ACO for distribution feeder reconfiguration, European Transactions on Electrical Power 20 (5) (2010) 575–590.
[38] T. Niknam, B. Amiri, An efficient hybrid approach based on PSO, ACO and *K*-means for cluster analysis, Applied Soft Computing 10 (1) (2010) 183–197.
[39] T.W. Liao, R.J. Kuo, J.T.L. Hu, Hybrid ant colony optimization algorithms for mixed discrete-continuous optimization problems, Applied Mathematics and Computation 219 (6) (2012) 3241–3252.
[40] J. Han, M. Kamber, Data Mining Concepts and Techniques, 2nd ed., Morgan Kaufmann Publishers, San Francisco, USA, 2006.
[41] J. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, L. Chretien, The dynamics of collective sorting robot-like ants and ant-like robots, in: Proceedings of the 1st International Conference on Simulation of Adaptive Behavior: From Animals to Animats, Paris, France, 1991, pp. 356–363.
[42] E. Lumer, B. Faieta, Diversity and adaptation in populations of clustering ants, in: Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3, Brighton, UK, 1994, pp. 501–508.
[43] N. Monmarche, M. Slimane, G. Venturini, AntClass: discovery of clusters in numeric data by a hybridization of an ant colony with the Kmeans algorithm, Internal Report No. 213, E3i, Universite de Tours, 1999, pp. 1–21.
[44] J. Handl, B. Meyer, Improved ant-based clustering and sorting in a document retrieval interface, in: Proceedings of the 7th International Conference on Parallel Problem Solving from Nature, Granada, Spain, 2002, pp. 913–992.
[45] A.L. Vizine, L.N. de Castro, E.R. Hruschka, R.R. Gudwin, Towards improving clustering ants: an adaptive ant clustering algorithm, Informatica 29 (2005) 143–154.
[46] U. Boryczka, Finding groups in data: cluster analysis with ants, Applied Soft Computing 9 (1) (2009) 61–70.
[47] P.S. Shelorar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Analytica Chimica Acta 509 (2) (2004) 187–196.
[48] M. Omran, A.P. Engelbrecht, A. Salman, Particle swarm optimization method for image clustering, International Journal of Pattern Recognition and Artificial Intelligence 19 (3) (2005) 297–322.
[49] D.W. Merwe, A.P. Engelbrecht, Data clustering using particle swarm optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, Canberra, Australia, 2003, pp. 215–220.
[50] X. Cui, T.E. Potok, Document clustering analysis based on hybrid PSO + *K*-means algorithm, Journal of Computer Science 1 (3) (2005) 27–33.
[51] P. Zhenkui, H. Xia, H. Jinfeng, The clustering algorithm based on particle swarm optimization algorithm, in: Proceedings of International Conference on Intelligent Computation Technology and Automation, Changsha, Hunan, China, 2008, pp. 148–151.
[52] A. Ahmadyfard, H. Modares, Combining PSO and *K*-means to enhance data clustering, in: Proceedings of the 4th IEEE International Symposium on Telecommunications, Tehran, Iran, 2008, pp. 688–691.
[53] C.Y. Tsai, I.W. Kao, Particle swarm optimization with selective particle regeneration for data clustering, Expert Systems with Applications 38 (6) (2011) 6565–6576.
[54] W. Conover, Practical Nonparametric Statistics, 2nd ed., Wiley & Sons Ltd., New York, 1980.
[55] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3–18.
[56] Y. Cooren, M. Clerc, P. Siarry, Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm, Swarm Intelligence 3 (2) (2009) 149–178.
[57] R. Tanese, Distributed genetic algorithms, in: Proceedings of the 3rd International Conference on Genetic Algorithms, San Francisco, CA, USA, 1989, pp. 434–439.
[58] C. Twomey, T. Stutzle, M. Dorigo, M. Manfrin, M. Birattari, An analysis of communication policies for homogeneous multi-colony ACO algorithms, Information Sciences 180 (12) (2010) 2390–2404.
[59] A. García-Villoria, R. Pastor, Introducing dynamic diversity into a discrete particle swarm optimization, Computers & Operations Research 36 (3) (2009) 951–966.