Investigating Data Clustering Using Hybridization Strategies of Continuous Ant Colony Optimization and Particle Swarm Optimization

Moein Fazeli Hassan Abadi*, Alireza Hamidi², Saeed Piry³, Massod Tezeroz 4

1M.Sc Student in Computer Science, University of Sistan and Baluchestan
2M.Sc Student in Software engineering, Islamic Azad University of Zahedan
3M.Sc Student in Software engineering, Islamic Azad University of Zahedan
4M.Sc Student in Software engineering, Islamic Azad University of Zahedan

*Corresponding Author: Moein Fazeli Hassan Abadi

Abstract

Clustering is involved with finding a structure in a collection of unlabeled data. Cluster is set of data that have similar. In the clustering attempts to divide data into clusters that similarity between data within each cluster, the maximum and minimum similarity between data in different clusters. Clustering techniques have received attention in many fields of study such as engineering, medicine, biology and data mining. In fact clustering Means is unsupervised data segmentation. Swarm intelligence algorithms including ant colony optimization (ACOR) and particle swarm optimization (PSO) have been applied to clustering in recent years. 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. This study incorporated and investigated ACOR with PSO for data clustering to improve the search ability. The Hybrid Models is tested on several data sets and its performance is compared with those of ACOR and PSO. The simulation results show that the proposed evolutionary optimization algorithm is robust and suitable for handling data clustering.

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

Introduction

Swarm intelligence (SI) originated from the study of colonies or swarms of social organisms (Engelbrecht, 2007). Among various algorithms in swarm intelligence, the ant colony algorithms (ACO) and particle swarm optimization (PSO) are two commonly used metaheuristic algorithms. The ant-based algorithm, chiefly used to solve combinatorial optimization problems, was inspired by observations of the foraging behavior of real ants (Dorigo, 1997; Bonabeau, 2000). 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 (Huang, 2013; Zhang, 2010). This approach does not generally suffice in accuracy because the length of the discretization interval affects the quality of solutions. Recently, Socha and Dorigo (Socha, 2008) proposed a new ant-based algorithm named ACOR to solve continuous optimization problems. PSO is inspired by social behavior among individuals, for instance, bird flocks. Particles (individuals) representing a potential problem solution move through the search space (Eberhart, 1995).

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. Premature convergence that leads to a fall into local optimum may exist in metaheuristic algorithms including ACOR and PSO. 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 (8,9). Hybridization in swarm intelligence is essential for performance enhancement of a swarm intelligence optimization algorithm. Swarm intelligence algorithms including ant colony optimization and particle swarm optimization have been applied to clustering in recent years. To date, numerous studies have applied data clustering by employing ACO or PSO (Kennedy, 2001; Dorigo, 2004). This study investigates several approaches for the combination of the ACOR and PSO for data clustering. 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.
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. Ant system utilizes a graph representation which augmented as follows: in addition to the cost measure \( \delta (r,s) \), each edge \((r,s)\) has also a desirability measure \( \tau (r,s) \), called pheromone, which is updated at run time by artificial ants. 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. The state transition rule used by ant system, called a random-proportional rule, is given by Eq. (1), which gives the probability with which ant \( k \) in city \( r \) chooses to move to the city \( s \).

\[
P_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^\beta [\mu(r,s)]^{\beta}}{\sum_{u \in J_k(r)} [\tau(u,s)]^\beta [\mu(u,s)]^{\beta}} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases}
\]

Where \( \tau \) is the pheromone, \( \mu = 1/\delta \) is the inverse of the distance \( \delta (r,s) \), \( J_k(r) \) is the set of cities that remain to be visited by ant \( k \) positioned on city \( r \) (to make the solution feasible), and \( \beta \) is a parameter which determines the relative importance of pheromone versus distance (\( \beta > 0 \)). In Eq. (1) we multiply the pheromone on edge \((r,s)\) by the corresponding heuristic value \( \mu (r,s) \). In this way we favor the choice of edges which are shorter and which have a greater amount of pheromone. In ant system, the global updating rule is implemented as follows. Once all ants have built their tours, pheromone is updated on all edges according to:

\[
\tau(r,s) = (1-\alpha) \tau(r,s) + \sum_{k=1}^{m} \Delta \tau_k(r,s)
\]

Where \( \Delta \tau_k(r,s) \) is calculated as follows:

\[
\Delta \tau_k(r,s) = \begin{cases} \frac{1}{L_k} & \text{if } (r,s) \in \text{tour done by ant } k \\ 0 & \text{otherwise} \end{cases}
\]

\( 0<\alpha<1 \) is a pheromone decay parameter, \( L_k \) is the length of the tour performed by ant \( k \), and \( m \) is the number of ants. Pheromone updating is intended to allocate a greater amount of pheromone to shorter tours. In a sense, this is similar to a reinforcement learning scheme (Barto 1981, Kaelbling, 2009) in which better solutions get a higher reinforcement (as happens, for example, in genetic algorithms under proportional selection). The pheromone updating formula was meant to simulate the change in the amount of pheromone due to both the addition of new pheromone deposited by ants on the visited edges, and to pheromone evaporation.

Ant algorithm for continuous domain

Socha and Dorigo (Socha, 2008) 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 (Ashtiani, 2011). 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: For each solution \( s_j \) in the pheromone table, calculate the value of the objective function \( f(s_j) \). Sort the solutions in the pheromone table according to their objective values, that is, for a minimum problem:

\[
f(s_1) \leq f(s_2) \leq \cdots \leq f(s_t) \leq \cdots \leq f(s_k)
\]

Where \( K \) is number of rows in the pheromone table. Calculate the weight \( w \) for each row in the pheromone table.

\[
w_i = \frac{1}{qK\sqrt{2\pi}} e^{-\frac{(i-1)^2}{2q^2K^2}}
\]

Where \( q \) represents the learning rate between 0 and 1. Compute the probability of the roulette wheel \( P_i \), according to the value of \( w_i \) for each solution:

\[
p_i = \frac{w_i}{\sum_{j=1}^{K} w_j}
\]

Repeat the following steps \( M \) times to generate \( M \) new ants: 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 \( \mu_i^{(d)} \) is a value selected from the \( d \)th value (variable) of the \( i \)th solution in the pheromone table by the probability of \( p_i \), and \( \sigma_i^{(d)} \) is defined as follows:
\[ a^d_t = \tau \sum_{j=1}^{k} \frac{|x^d_i - x^d_j|}{K-1} \]  

(5)

Where \( x^d_i \) is the value of the \( d \)th variable of the \( t \)th solution, \( k \) is the size of the pheromone table, and \( \tau \) represents the evaporation rate. Evaluate the M new ants and replace the inferior solutions in the pheromone table by the superior solutions from the M new ants.

**Particle swarm optimization**

Particle Swarm Optimization (PSO) Developed by Kennedy and Eberhart in 1995 (Eberhart, 1995), which is a population based optimization technique inspired by social behavior of bird flocking or fish schooling. PSO consists of a swarm of particles. Each particle resides at a position in the search space, The fitness of each particle represents the quality of its position, The particles fly over the search space with a certain velocity, The velocity (both direction and speed) of each particle is influenced by its own best position found so far and the best solution that was found so far by its neighbors, Eventually the swarm will converge to optimal positions. The detail algorithm of PSO is as follows:

- Randomly initialize particle positions and velocities
- While not terminate
  - For each particle \( i \):
    - Evaluate fitness \( y_i \) at current position \( x_i \)
    - If \( y_i \) is better than \( pbest_i \) then update \( pbest_i \) and \( p_i \)
    - If \( y_i \) is better than \( gbest_i \) then update \( gbest_i \) and \( g_i \)
  - For each particle
    - Update velocity \( v_i \) and position \( x_i \) using:
      \[
      \begin{align*}
      v_i &= v_i + u(0, \phi_1) \times (p_i - x_i) + u(0, \phi_2) \times (g_i - x_i) \\
      x_i &= x_i + v_i
      \end{align*}
      \]  
      (6)

For each particle \( i \), \( x_i \) is a vector denoting its position, \( v_i \) is the vector denoting its velocity, \( y_i \) denotes the fitness score of \( x_i \), \( p_i \) is the best position that it has found so far, \( pbest_i \) denotes the fitness of \( p_i \), \( g_i \) is the best position that has been found so far in its neighborhood, \( gbest_i \) denotes the fitness of \( g_i \), \( U(0, \phi) \) is a random vector uniformly distributed in \([0, \phi]\) generated at each generation for each particle, \( \phi_1 \) and \( \phi_2 \) are the acceleration coefficients determining the scale of the forces in the direction of \( p_i \) and \( g_i \).

**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 (Han, 2006). Swarm intelligence, including ant algorithms and PSO, has been used in clustering. Ant-based clustering was first introduced by Deneubourg et al (Deneubourg, 1991). Several studies have applied PSO in clustering (Omran, 2005; Tsai, 2011). These literatures showed that with certain enhancements, PSO can avoid trapping into local optimum, and outperformed a few other traditional clustering algorithms.

**Hybridization Strategies of ACOR and PSO**

In 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:

\[
F = \sum_{k=1}^{N_c} \sum_{i=1}^{N_i} \min ||x_i - c_k||^2 \sum_{k=1}^{N_c} d(C_k, C_j)
\]  

(7)

\( N_c \) = number of centers
\( N_i \) = sample size
\( \min ||x_i - c_k||^2 \) = distance between sample \( i \) to center \( k \)
\( d(C_k, C_j) \) = distance between center \( i \) and center \( k \)

**Sequential approach**

In the sequential approach, the ACOR and PSO share the same set of solutions, named the "pheromone-particle" table, which this set of solutions include pheromone in ACOR and the current solution 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. Figure 1 shows the main steps of the sequential approach (Huang, 2013).

**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. Figure 2 shows the main steps of the parallel approach (Huang, 2013).

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

Based on the pheromone-particle table, the PSO generates K new particles, which combined with the pheromone-particle table to form an enlarged pheromone-particle table with the size of 2K. 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. Figure 3 shows the main steps of the sequential approach with the enlarged pheromone-particle table (Huang, 2013).
Global best exchange

In the global best exchange, 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. Figure 4 shows the main steps of the Global best exchange approach (Huang, 2013).

Experiments

Four hybrid model, standalone PSO and standalone ACOR tested for clustering with two dataset include Iris dataset and Wine dataset from UCI dataset, table 1 show datasets from the UCI repository. The clustering performances are calculated by using Eq 6. table 2 show average 5-run performances using Iris dataset for PSO, ACOR and four types of hybrid models and table 3 show average 5-run performances using Wine dataset for PSO, ACOR and four types of hybrid models. 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 1. Average 5-run performances using Iris dataset for PSO, ACOR and four types of hybrid models.

<table>
<thead>
<tr>
<th>Evolutionary Algorithms</th>
<th>The average cost of the best solution in five-run</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>14.1916</td>
</tr>
<tr>
<td>ACOR I</td>
<td>13.0434</td>
</tr>
<tr>
<td>Hybrid I</td>
<td>9.0470</td>
</tr>
<tr>
<td>Hybrid II</td>
<td>10.2085</td>
</tr>
<tr>
<td>Hybrid III</td>
<td>8.7959</td>
</tr>
<tr>
<td>Hybrid IV</td>
<td>9.4821</td>
</tr>
</tbody>
</table>

Table 2. Average 5-run performances using Wine dataset for PSO, ACOR and four types of hybrid models.

<table>
<thead>
<tr>
<th>Evolutionary Algorithms</th>
<th>The average cost of the best solution in five-run</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>13.5010</td>
</tr>
<tr>
<td>ACOR</td>
<td>12.5512</td>
</tr>
<tr>
<td>Hybrid I</td>
<td>10.9316</td>
</tr>
<tr>
<td>Hybrid II</td>
<td>11.2854</td>
</tr>
<tr>
<td>Hybrid III</td>
<td>10.1040</td>
</tr>
<tr>
<td>Hybrid IV</td>
<td>11.5279</td>
</tr>
</tbody>
</table>

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. The experimental results also show the ACOR performance is superior to that of PSO, and the performance of Hybrid models superior to that of the PSO and ACOR.

References


