Clustering by Ant Colony Optimization

Chapter · January 2004

DOI: 10.1007/978-3-642-17103-1_3

Javier Trejos
University of Costa Rica

Eduardo Piza
University of Costa Rica

Clustering by Ant Colony Optimization

Javier Trejos[1], Alex Murillo[2], and Eduardo Piza[3]

[1] CIMPA, School of Mathematics, University of Costa Rica, 2060 San José, Costa Rica. jtrejos@cariari.ucr.ac.cr
[2] Department of Statistics and Operations Research, University of Granada, 18071 Granada, Spain. alexmurillo65@hotmail.com
[3] CIMPA, School of Mathematics, University of Costa Rica, 2060 San José, Costa Rica. epiza@cariari.ucr.ac.cr

Summary. We use the heuristics known as ant colony optimization in the partitioning problem for improving solutions of k-means method. Each ant in the algorithm is associated to a partition, which is modified by the principles of the heuristics; that is, by the random selection of an element, and the assignment of another element which is chosen according to a probability that depends on the pheromone trail (related to the overall criterion: the maximization of the between-classes variance), and a local criterion (the distance between objects). The pheromone trail is reinforced for those objects that belong to the same class. We present some preliminary results, compared to results of other techniques, such as simulated annealing, genetic algorithm, tabu search and k-means. Results are as good as the best of the above methods.

1 Introduction

In the partitioning problem it is well known the phenomenon of local minima. Indeed, most of the methods for clustering with a fixed number of classes find local optima of the criterion to be optimized, such as k-means or its variants (dynamical clusters, transfers, Isodata [DID82, BOC74]). This is the reason why the implementation of modern combinatorial optimization heuristics has been studied, such as simulated annealing [AAR88], tabu seach [GLO93] and genetic algorithms [GOL89], which have shown to have good features when implemented in different problems. These heuristics have been widely studied by the authors in numerical clustering [TRE98], binary clustering [TRE01] or two-mode clustering [TRE00, CAS02], with good results.

In this article we study a recent optimization heuristic, the *ant colony optimization* (ACO) [DOR99], which has shown to have good performances in some well known problems of operations research. Section 2 reminds us of the partitioning problem that is considered here: a data table with numerical data and criterion to be optimized. Section 3 presents the technique and takes a closer look at ACO with the help of two problems where it has found good results: the traveling salesman problem and the quadratic assignment problem. In Sect. 4 we present the algorithm using ACO for the minimization of the intra-classes variance: an algorithm for the modification of partitions by the transfer of objects according to a probabilistic rule based on a pheromone trail of reinforcement and a local criterion of closeness of objects. Finally, in Sect. 5 we report the obtained results on some real data tables.

2 Partitioning

We are in the presence of a data set $\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of objects in $\mathbb{R}^p$, and we look for a partition $P = (C_1, \ldots, C_K)$ of Ω that minimizes the intra-classes variance

$$W(P) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \omega_i \, \|\mathbf{x}_i - \mathbf{g}_k\|^2 \tag{1}$$

where $\mathbf{g}_k$ is the gravity center or mean vector of class C_k and ω_i is the weight of object $\mathbf{x}_i$; in the simplest case, $\omega_i = 1/n$. It is well known that the minimization of $W(P)$ is equivalent to the maximization of the inter-classes variance

$$B(P) = \sum_{k=1}^{K} \mu_k \, \|\mathbf{g}_k - \mathbf{g}\|^2 \tag{2}$$

where $\mu_k = \sum_{\mathbf{x}_i \in C_k} \omega_i$ is the class weight of C_k, and $\mathbf{g} = \sum_{i=1}^{n} \omega_i \, \mathbf{x}_i$ is the overall gravity center, since $I = W(P) + B(P)$, $I = \sum_{i=1}^{n} \omega_i \, \|\mathbf{x}_i - \mathbf{g}\|^2$ being the total variance of Ω.

The most widely known method of partitioning is k-means, which is a local seach method; the final solution depends deterministicly on the initial partition. The application of modern optimization heuristics [TRE98, PIZ95, MUR00] —such as simulated annealing, tabu search and genetic algorithms— is based on the transfer of objects between classes. In the case of **simulated annealing** we use the Metropolis rule for deciding whether a transfer of an object from a class to another one —both selected at random— is accepted. In **tabu search**, we construct a set of partitions (the neighborhood of a given partition) by the transfer of a single object and the best neighbor is chosen accordingly to the rules of this technique. Finally, we apply a **genetic algorithm** with a chromosomic representation of n allele in an alphabet of K letters that represent a partition, and we use a roulette-wheel selection proportional to $B(P)$, mutations (that correspond to transfers), and a special crossover operator called "forced crossover" (a good father imposes membership in a class —chosen at random— to another father). In all three cases, the use of the heuristics showed a clearly better performance than k-means or Ward hierarchical clustering. We have also adapted these heuristics to binary data, making the corresponding changes to the dissimilarities, distances and aggregation indexes in that context, since usually there is no “centroid” for binary data (except for the L_1 distance).

3 Ant Colony Optimization

The underlying metaphor of ant colony optimization (ACO) is the way that some insects living in collaborative colonies look for food. Indeed, if an ant nest feels a food source, then some expeditions of ants go —by different paths— to search for this food, leaving a pheromone trail, a chemical substance that animals usually have, but very important for insects. This pheromone trail is an olfative signal for other ants, that will recognize the way followed by its predecessors. Between all expeditions of ants, there will be some that arrive first to the food source because they took the shortest path, and then they will go back to the nest first than the other expeditions. Then, the shortest path has been reinforced in its pheromone trail; therefore, new expeditions will probably take that path more than others will, unless new better paths (or parts of paths) are found by some expeditions. It is expected that the pheromone trail of the shortest path is more and more intense, and the one of the other paths will evaporate.

When applying this principle to combinatorial optimization problems, we look for an implementation that uses the principle of *reinforcement of good solutions*, or parts of solutions, by the intensification of a value of “pheromone” that controls the probability of taking this solution or its part of solution. Now, this probability will depend not only on the pheromone value, but also on a value of a “local heuristic” or “short term vision”, that suggests a solution or part of solution by a local optimization criterion, for example as greedy algorithms do.

An optimization method that uses ACO has at least the following components:

- A *representation*, that enables the construction or modification of solutions by means of a probabilistic transition rule, that depends on the pheromone trail and the local heuristic.
- A *local heuristic* or *visibility*, noted η.
- An *update rule* for the pheromone, noted τ.
- A *probabilistic transition rule*, that depends on η and τ.

One of the principles of ACO is that it handles in parallel a set of M agents or ants, and each one constructs or modifies a solution of the optimization problem.

The general algorithm of ACO, in the case that states are vertices $i, j, \ldots$ on a graph of n nodes, is the following:

Algorithm ACO
Initialize $\tau_{ij} = \tau_0$
Put each ant (from 1 to M) in a vertex
for $t = 1$ to $t_{\max}$ do:
  for $m = 1$ to M do:
    Construct a solution $S^m(t)$ applying $n-1$ times a rule
    of construction or modification, choosing
    a pheromone trail τ and a local heuristics η
    Calculate the cost $W^m(t)$ of $S^m(t)$
  end-for
  for each arc (i,j) do:
    Update τ
  end-for
end-for.

We briefly explain how ACO has been implement in the travelling salesman problem (TSP) and the quadratic assignment problem (QAP).

In the TSP, each ant constructs a permutation; at each step, the m-th ant is on a city i and selects the next city j with probability $p_{ij}^m = [\tau_{ij}(t)]^\alpha[\eta_{ij}]^\beta / \sum_{h\in J_i^m(t)}[\tau_{ih}(t)]^\alpha[\eta_{ih}]^\beta$, where $J_i^m(t)$ is the set of cities that can be visited by m when it is on i, and t is the current iteration. The local heuristics is the inverse of the distance: $\eta_{ij} = 1/d_{ij}$. Many variants have been proposed [DOR99], for example the use of elite states for updating the pheromone value. Here $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\sum_{m=1}^M \Delta^m\tau_{ij}(t+1)$ where $i,j \in \{1,\dots,n\}$, $\rho \in\,]0,1]$, and

$$\Delta^m\tau_{ij}(t+1) = \begin{cases} Q/L_k & \text{if ant } m \text{ uses arc } (i,j) \text{ in its permutation} \\ 0 & \text{otherwise,} \end{cases}$$

is the amount of pheromone left by ant m, Q being a constant and L_k the length of the cycle constructed by m.

For the QAP, there is also a constructive algorithm very similar to the TSP one, with a local heuristic $\eta_{ih} = d_i f_h$, where $d_i = \sum_{j=1}^n d_{ij}$ and $f_h = \sum_{k=1}^n f_{hk}$ is the sum of activities flux. There is also a modifying algorithm for QAP [GAM99], where each ant is associated to a permutation π^m (initialized at random and improved by a local search method). The ant modifies π^m by means of R swaps, $\pi^m(i) \leftrightarrow \pi^m(j)$. With probability $q \in\,]0,1[$, j is chosen such that $\tau_{i\pi^m(j)} + \tau_{j\pi^m(i)}$ is maximized, and with probability $1-q$ it is chosen j with probability $p_{ij}^m = \tau_{i\pi^m(j)} + \tau_{j\pi^m(i)} / \sum_{\substack{l=1 \\ l\neq i}}^n \tau_{i\pi^m(l)} + \tau_{l\pi^m(i)}$. The pheromone trail $\tau_{i\pi(i)}$ is updated by $\Delta\tau_{ij}(t) = 1/C(\pi^+)$ if (i,j) belongs to the best state π^+ found so far; that is, there is an elite of one element. There are some other aspects, such as intensification and diversification that can be consulted in [GAM99].

4 Application of ACO in partitioning

We propose an iterative algorithm such that in each iteration the behavior of every ant is examined. At the beginning, an ant m is associated to a partition P^m randomly generated, the k-means method is applied and converges to a local minimum of W. During the iterations, the ant will modify P^m in the following way: an object i is selected at random, and another object j is selected at random using a roulette-wheel with probability p_{ij}, where p_{ij} depends on the pheromone trail and a local heuristic. We can say that the ant decides whether to assign j to the same class as i.

The value of the pheromone trail is modified according to the rule

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\,\Delta\tau_{ij}(t+1) \tag{3}$$

where the values τ_{ij} associate two objects i, j in Ω, and $\rho \in]0,1]$ is an *evaporation parameter*. We put

$$\Delta\tau_{ij}(t+1) = \sum_{m=1}^{M} \Delta^m\tau_{ij}(t+1), \tag{4}$$

$\Delta^m\tau_{ij}(t+1)$ being the quantity of pheromone by agent in the association of objects i, j to the same class, defined by

$$\Delta^m\tau_{ij}(t+1) = \begin{cases} B(P^m)/I & \text{if } i,j \text{ belong to the same class of } P^m \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

$B(P^m)$ being the inter-classes variance of partition P^m. That way, two objects classified into the same class leave a pheromone trail.

The local heuristic or short-term visibility is defined as

$$\eta_{ij} = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|}, \tag{6}$$

in such a way that two near elements give a big value in order to influence in the probability of assigning them to the same class.

If ant m is at object i, object j is chosen with probability

$$p_{ij} = \frac{[\tau_{ij}]^\alpha[\eta_{ij}]^\beta}{\sum_{l=1}^{n}[\tau_{il}]^\alpha[\eta_{il}]^\beta} \tag{7}$$

Then, j is assigned to the same class as i. This random choice is similar to the so called roulette-wheel of genetic algorithms: the rows of matrix $(p_{ij})_{n\times n}$ sum 1; given i, the value p_{ij} is the probability of choosing j, which is modeled using the cumulative probabilities and generating uniformly random numbers.

With the above elements, the algorithm is as follows:

Algorithm Acoclus
Initialize $\tau_{ij} = \tau_0$
Calculate η according to (6)
Initialize the probabilities p
Initialize at random partitions P^1,. . .,P^M associated to each ant
Run k-means on each P^m in order to converge to a local minimum of W
for $t = 1$ to $t_{\max}$ do:
  for $m = 1$ to M, do S times:
    Choose at random an object i
    Choose an object j according to (7)
    Assign j to the class of i
  end-for
  Calculate $B(P^1), \ldots, B(P^M)$ and keep the best value
  Update τ according to (3), (4) and (5)
end-for.

Notice that the algorithm has the following parameters: the number M of ants, the initial value of pheromone τ_0, the maximum number of iterations $t_{\max}$, the power α of pheromone in (7), the power β of the local heuristic in (7), the evaporation coefficient ρ in (3), and the number S of transfers for each ant.

5 Results and perspectives

As with other metaheuristics, in ant colony optimization there is a problem with the tuning of the parameters. In ACO this problem is more complicated, because it has more parameters than the other heuristics. In ACO we have 7 parameters to be fixed, instead of 4 in simulated annealing and 2 in tabu search.

Some preliminary experimentation gave us the following:

- If $\alpha, \beta > 1$, it is better to have bigger values of β;
- β has to be larger than α;
- $\alpha, \beta < 1$ is better than $\alpha, \beta \geq 1$;
- $\rho \approx 0.3$ is better than $\rho \approx 0.5$ or $\rho \approx 0.7$.

We also tried another rule for choosing j when i has been selected: with probability $q \in]0, 1[$ choose j that maximizes τ_{ij}: $j = \text{argmax}_h\{\tau_{ih}\}$, and with probability $1 - q$ choose j according to (7). However, this change did not give better results.

A significant improvement was obtained when k-means was run before starting the iterations, since at the beginning of our experimentation initial partitions were purely at random and ants improved them only during the iterations. So, we can see our method as a method for improving k-means solutions in partitioning.

With all the above considerations, the following results were obtained by the application of AcoClus with:

- $M = 20$: population size
- maximum number of iterations: 20
- number of runs: 25
- $S = 2 \times n$: number of transfers
- $\alpha = 1$
- $\beta = 0.8$
- $\rho = 0.3$
- $\tau_0 = 0.001$

The method is applied several times with the same parameters, and the attraction rate of the best minimum of W is examined. Table 1 shows the results of applying the methods based on combinatorial optimization heuristics and AcoClus (applied 25 times), on four data tables described by the literature [TRE98]. Tabu search was applied 150 times, while simulated annealing was applied 1,000 times on the first table, 200 times on the second and third, and 25 on the fourth; the genetic algorithm was applied 100 times on the first three tables, and 50 times on the fourth. The dimension $n \times p$ of each table appears close to the table's name. Only the best value of W is reported, and for each method we report the percentage of times this value was found. We also report the results for k-means (applied 10,000 times) and Ward hierarchical clustering (applied only once since it is deterministic).

Results on table 1 show that ACO performs well, as good as simulated annealing and tabu search. However, some experimentation has yet to be done for deciding some features of the method. For example, we have to decide whether to use elite agents or not, to "kill" the ants before restarting a new iteration or not, and the use of another rule for selecting j from i. A complete Monte Carlo comparison is being performed at the present time, controlling four factors: the data table size, the number of clusters, the cardinality and the variance of each class. The results should compare ant colony optimization, simulated annealing, tabu search, genetic algorithms, particle swarm optimization, k-means and Ward's hierarchical clustering.

References

[AAR88] Aarts, E.M., Korst, J.: Simulated Annealing and Boltzmann Machines. John Wiley & Sons, Chichester (1988)

[BOC74] Bock, H.-H.: Automatische Klassifikation. Vandenhoeck & Ruprecht, Göttingen (1974)

[DOR99] Bonabeau, E.: Dorigo, M., Therauluz, G.: Swarm Intelligence. From Natural to Artificial Systems. Oxford University Press, New York (1999)

[CAS02] Castillo, W., Trejos, J.: Two-mode partitioning: review of methods and application of tabu search. In: Jajuga, K. et al. (eds) Classification, Clustering, and Data Analysis, Springer, Berlin Heidelberg New York (2002)

Table 1. Best value of the intra-classes variance W^* and percentage of times this value is obtained when each method is applied several times: ant colony optimization (ACO), tabu search (TS), simulated annealing (SA), genetic algorithm (GA), k-means (kM), and Ward's hierarchical.

K	W^*	ACO	TS	SA	GA	KM	Ward
	French Scholar Notes (9×5)						
2	28.2	100%	100%	100%	100%	12%	0%
3	16.8	100%	100%	100%	95%	12%	0%
4	10.5	100%	100%	100%	97%	5%	100%
5	4.9	100%	100%	100%	100%	8%	100%
	Amiard's Fishes (23×16)						
3	32213	100%	100%	100%	87%	8%	0%
4	18281	100%	100%	100%	0%	9%	0%
5	14497	68%	97%	100%	0%	1%	100%
	Thomas' Sociomatrix (24×24)						
3	271.8	100%	100%	100%	85%	2%	0%
4	235.0	96%	100%	100%	24%	0.15%	0%
5	202.6	84%	98%	100%	0%	0.02%	0%
	Fisher's Iris (150×4)						
2	0.999	100%	100%	100%	100%	100%	0%
3	0.521	100%	76%	100%	100%	4%	0%
4	0.378	100%	60%	55%	82%	1%	0%
5	0.312	100%	32%	0%	6%	0.24%	0%

[DID82] Diday, E., Lemaire, J., Pouget, J., Testu, F.: Eléments d'Analyse des Données. Dunod, Paris (1982)

[GAM99] Gambardella, L.M., Taillard, E.D., Dorigo, M.: Ant colonies for the QAP. J. Oper. Res. Soc., **50**, 167–176 (1999)

[GLO93] Glover, F. et al.: Tabu search: an introduction. Annals of Operations Research, **41**, 1–28 (1993)

[GOL89] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading-Mass (1989)

[MUR00] Murillo, A.: Aplicación de la búsqueda tabú en la clasificación por particiones. Investigación Operacional, **21**, 183–194 (2000)

[PIZ95] Piza, E., Trejos, J.: Particionamiento usando sobrecalentamiento simulado y algoritmos genéticos. In: Trejos, J. (ed) IX Simposio Internacional de Métodos Matemáticos Aplicados a las Ciencias. UCR, Turrialba (1995)

[TRE98] Trejos, J., Murillo, A., Piza, E.: Global stochastic optimization for partitioning. In: Rizzi, A. et al. (eds) Advances in Data Science and Classification. Springer, Berlin Heidelberg New York (1998)

[TRE00] Trejos, J., Castillo, W.: Simulated annealing optimization for two-mode partitioning. In: Gaul, W. & Decker, R. (eds) Classification and Information Processing at the Turn of the Millenium. Springer, Berlin Heidelberg New York (2000)

[TRE01] Trejos, J., Piza, E.: Critères et heuristiques d'optimisation pour la classification de données binaires. In: Journées de la Société Francophone de Classification, Guadeloupe (2001)