Variants of Ant Colony Optimization: A Metaheuristic for Solving the Traveling Salesman Problem

Iván Chaparro and Fevrier Valdez

Tijuana Institute of Technology, Tijuana, México

Abstract. Ant Colony Optimization (ACO) has been used to solve several optimization problems. However, in this paper, the variants of ACO have been applied to solve the Traveling Salesman Problem (TSP), which is used to evaluate the variants ACO as Benchmark problems. Also, we developed a graphical interface to allow the user input parameters and having as objective to reduce processing time through a parallel implementation. We are using ACO because for TSP is easily applied and understandable. In this paper we used the following variants of ACO: Max-Min Ant System (MMAS) and Ant Colony System (ACS).

Keywords: ACO, TSP, Optimization, Combinatorial Problems.

1 Introduction

There are different algorithms based on the simulation of natural processes and genetics such as genetic algorithms and Ant Colony Optimization (ACO), based on heuristic problem solving. Currently required to solve more complex problems which require too much processing time for result. Therefore, we can work with highly complex problems getting results with less processing time with a parallel implementation. In this paper we describe several variants of Ant Colony Optimization (ACO) to solve the Traveling Salesman Problem (TSP) allowing the user input parameters using a graphical interface and performing parallel processing.

2 ACO Variants

The ACO metaheuristic is inspired by observing the behavior of real ant colonies, which presented as an interesting feature how to find the shortest paths between the nest and food, on its way the ants deposit a substance called pheromone, this trail allows the ants back to their nest from the food; it uses the evaporation of pheromone to avoid a unlimited increase of pheromone trails and allow to forget the bad decisions, thus avoiding the persistence of the pheromone trails and therefore, the stagnation in local optima [4].

O. Castillo et al. (Eds.): Recent Advances on Hybrid Intelligent Systems, SCI 451, pp. 323–331.
springerlink.com

2.1 Traveling Salesman Problem (TSP)

This problem is defined as to visit "n" cities, starting and ending with the same city, visiting each city once and making the tour with the lowest cost, this cost can be expressed in terms of time or distance, i.e., travel a minimum of kilometers or perform a tour in the shortest time possible. More formally, the TSP can be represented by a complete weighted graph G= (N, A) with N being the set of nodes representing the cities, and A being the set of arcs. Each arc (i, j) ∈ A is assigned a value (length) d_{ij}, which is the distance between cities i and j, with i,j ∈ N. In the general case of the asymmetric TSP, the distance between a pair of nodes i,j is dependent on the direction of traversing the arc, that is, there is at least one arc (i, j) for $d_{ij} \neq d_{ji}$. In the symmetric TSP, $d_{ij} = d_{ji}$, holds for all the arcs in A. The goal in the TSP is find a minimum length Hamiltonian circuit of the graph, where a Hamiltonian circuit is a closed path visiting each of n=$|N|$ nodes of G exactly once. Thus, an optimal solution to the TSP is a permutation π of the node indices {1,2,…,n} such that the length $f(\pi)$ is minimal, where $f(\pi)$ is given by[4]:

$$f(\pi) = \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)} \quad (1)$$

2.2 Max-Min Ant System

This algorithm introduces four main modifications with respect to the Ant System [3]:

- It strongly exploits the best tours found.
- It limits the possible range of pheromone trail values to the interval $[\tau_{min}\ \tau_{max}]$.
- The pheromone trails are initialized to the upper pheromone trail limit, which, together a small pheromone evaporation rate, increases the exploration of tours at the start of the search.
- Pheromone trails are reinitialized each time the system approaches stagnation or when no improved tour has been generated for a certain numbers of consecutive iterations.

2.3 Ant Colony System (ACS)

This algorithm was proposed by Dorigo and Gambardella in the year 1997, and differs from Ant System in three main points [4]:

- It exploits the search experience accumulated by the ants more strongly than Ant System does through the use of a more aggressive action choice rule.

- Pheromone evaporation and pheromone deposit take a place only on the arcs belonging to the best-so-far tour.
- Each time an ant uses an arc (i,j) to move from city i to city j, it removes some pheromone from the arc to increase the exploration of alternative paths.

3 Graphical Interface in Matlab

A graphical Interface was developed with the objective of introducing parameters for the variants of ACO. The first user interface was made only as presentation to enter the options menu as shown in the figure 1.

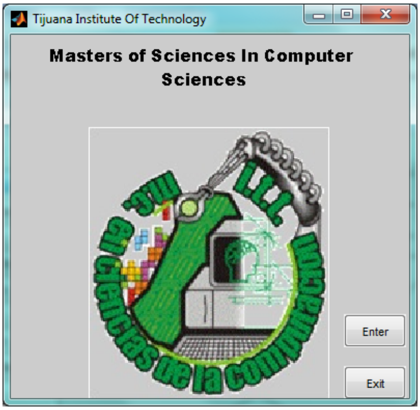

Fig. 1. Main Interface

Or to exit press the button exit, showing the following confirmation message:

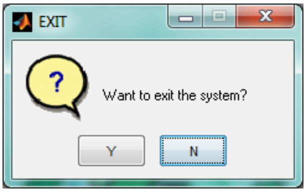

Fig. 2. Interface confirmation message

Figure 3, select the type of variant and method of execution or press the back button to go to the previous interface.

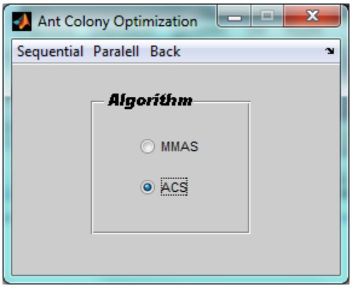

Fig. 3. Menu of options

Figure 4, show the execution time and minimum distance that is the cost of taking the tour of cities in addition to plot the location of cities

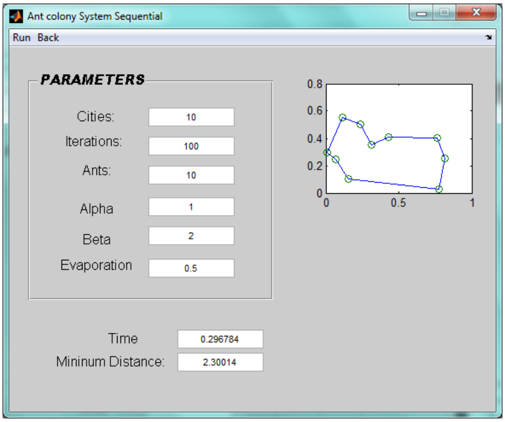

Fig. 4. Interface to introduce parameters

4 Results

This section shows the results obtained from experiments with the different variants of ACO with sequential and parallel processing.

30 experiments were performed where vary alpha, beta, rho factor and the number of iterations using 10 cities, the results are shown in table 1, we obtained a minimum distance of 2.3001 with 100 and 7 iterations of the algorithm; however, for less generations the processing time is lower reducing the execution time to 0.113762 seconds in experiment number 26. Average was obtained by execution of 1.48 seconds. An experiment was performed only with 1000 cities for more complexity to the algorithm and obtained an execution time of 16 seconds.

Table 1. Sequential Variant for ACS

EXPERIMENT	CITIES	ANTS	ITERATIONS	ALPHA	BETA	RHO	MINIMUN DISTANCE	TIME
1	10	100	100	1	2	0.3	2.4064	**2.297667**
2	10	100	100	1	2	0.3	2.4064	**2.204522**
3	10	100	100	1	2	0.3	2.4064	**2.151696**
4	10	100	100	1	2	0.3	2.4064	**2.191324**
5	10	100	100	1	2	0.3	2.4064	**2.1652**
6	10	100	100	1	2	0.3	2.4064	**2.185889**
7	10	100	100	1	2	0.3	2.4064	**2.171482**
8	10	100	100	1	2	0.3	2.4064	**2.172033**
9	10	100	100	1	2	0.3	2.4064	**2.175278**
10	10	100	100	1	2	0.3	2.4064	**2.379699**
11	10	100	100	2	3	0.5	2.3001	**2.293828**
12	10	100	100	2	3	0.5	2.3001	**2.343802**
13	10	100	100	2	3	0.5	2.3001	**2.364691**
14	10	100	100	2	3	0.5	2.3001	**2.3829**
15	10	100	100	2	3	0.5	2.3001	**2.330182**
16	10	100	100	2	3	0.5	2.3001	**2.277928**
17	10	100	100	2	3	0.5	2.3001	**2.261657**
18	10	100	100	2	3	0.5	2.3001	**2.288932**
19	10	100	100	2	3	0.5	2.3001	**2.321291**
20	10	100	10	2	3	0.5	2.3001	**0.301492**
21	10	100	10	2	3	0.5	2.3001	**0.266089**
22	10	100	10	2	3	0.5	2.3001	**0.269683**
23	10	10	10	2	3	0.5	2.3001	**0.072738**
24	10	10	9	2	3	0.5	2.3001	**0.125904**
25	10	10	8	2	3	0.5	2.3001	**0.078677**
26	10	10	7	2	3	0.5	2.3001	**0.113762**
27	10	10	6	2	3	0.5	2.5144	**0.075725**
28	10	10	5	2	3	0.5	2.5319	**0.081266**
29	10	10	4	2	3	0.5	2.4064	**0.07106**
30	10	10	3	2	3	0.5	2.53.19	**0.078017**
31	1000	100	10	2	3	0.5	40.4873	997.667909
							1.483147133	**Seconds**
							16.62779848	**Seconds**

The same numbers of experiments were performed using the same parameters for parallel execution, finding the minimum distance of 2.3001 also with 7 iterations but with less processing time of 0.079525 second in experiment number 26. In the parallel implementation are executed 4 processes simultaneously, therefore, the average by execution of one process is 0.01988125 seconds and obtained an average of all experiments of 1.05 seconds and obtained 8.7125918 seconds for the experiment was done with 1000.

MMAS variant to obtain the minimum distance of 46.5519 in all experiments and only 2 iterations is obtained the same distance showing the result in table 3.Average was obtained by execution of all experiments of 5.81 seconds and performed an experiment with a 22 cities with a time of 20.329047 seconds.

In parallel the same distance was obtained only reducing the processing time. The results are shown in Table 4.

Table 2. Parallel Variant for ACS

EXPERIMENT	CITIES	ANTS	ITERATIONS	ALPHA	BETA	RHO	MINIMUN DISTANCE	lab1	lab2	lab3	lab4
1	10	100	100	1	2	0.3	2.3001	4.985979	4.670103	**5.803356**	4.450128
2	10	100	100	1	2	0.3	2.3001	4.717038	**5.766149**	4.855537	4.638474
3	10	100	100	1	2	0.3	2.3001	**5.663317**	4.626035	4.773888	4.315661
4	10	100	100	1	2	0.3	2.3001	4.76076	4.650837	**5.480135**	4.432192
5	10	100	100	1	2	0.3	2.3001	4.562846	4.682623	4.794606	**5.590443**
6	10	100	100	1	2	0.3	2.3001	4.793052	4.706704	**5.581461**	4.903088
7	10	100	100	1	2	0.3	2.3001	4.675758	4.627855	**5.557725**	4.282445
8	10	100	100	1	2	0.3	2.3001	5.676394	4.586213	**5.067412**	4.835595
9	10	100	100	1	2	0.3	2.3001	4.690962	**5.544794**	4.667791	4.765641
10	10	100	100	1	2	0.3	2.3001	4.640137	4.616093	**5.856676**	4.217882
11	10	100	100	2	2	0.5	2.3001	4.740937	4.697812	4.502633	**5.72732**
12	10	100	100	2	2	0.5	2.3001	4.509803	4.65397	4.752962	**5.862888**
13	10	100	100	2	2	0.5	2.3001	5.163928	4.935564	4.534519	**5.81736**
14	10	100	100	2	2	0.5	2.3001	5.184917	**5.842456**	4.668559	4.347392
15	10	100	100	2	2	0.5	2.3001	4.7837	4.693819	4.962278	**6.054081**
16	10	100	100	2	2	0.5	2.3001	4.615381	**5.656933**	4.580497	4.195778
17	10	100	100	2	2	0.5	2.3001	**5.58897**	4.735533	5.434506	4.359453
18	10	100	100	2	2	0.5	2.3001	5.051927	4.583253	**5.678237**	4.344494
19	10	100	100	2	2	0.5	2.3001	4.30736	4.675053	4.67073	**5.894579**
20	10	100	100	2	2	0.5	2.3001	4.667901	4.375274	4.707069	**5.797504**
21	10	100	100	2	2	0.5	2.3001	**5.730341**	4.426355	4.604745	4.909657
22	10	100	100	2	2	0.5	2.3001	4.583651	4.664926	**5.881932**	4.22536
23	10	10	10	2	2	0.5	2.3001	0.070359	**0.077827**	0.070337	0.05964
24	10	10	9	2	2	0.5	2.3001	**0.108555**	0.072157	0.072857	0.05581
25	10	10	8	2	2	0.5	2.3001	**0.096793**	0.065903	0.045364	0.072012
26	10	10	7	2	2	0.5	2.3001	0.062876	0.058779	**0.079525**	0.045117
27	10	10	6	2	2	0.5	2.5144	0.052096	0.04201	**0.106847**	0.042266
28	10	10	5	2	2	0.5	2.5319	0.040768	**0.061523**	0.03778	0.052522
29	10	10	4	2	2	0.5	2.4064	**0.132485**	0.060001	0.047999	0.046622
30	10	10	3	2	2	0.5	2.5319	**0.059991**	0.044205	0.050919	0.032897
31	1000	100	10	2	2	0.5	40.4873	1038.578504	1045.511016		
								1.051396792	**Seconds**		
								8.7125918	**Seconds**		

Table 3. Sequential Variant for MMAS

EXPERIMENT	CITIES	ANTS	ITERATIONS	ALPHA	BETA	RHO	MINIMUN DISTANCE	TIME
1	10	100	100	1	2	0.3	46.5519	**8.513724**
2	10	100	100	1	2	0.3	46.5519	**8.540868**
3	10	100	100	1	2	0.3	46.5519	**8.561641**
4	10	100	100	1	2	0.3	46.5519	**8.786021**
5	10	100	100	1	2	0.3	46.5519	**8.463686**
6	10	100	100	1	2	0.3	46.5519	**8.513592**
7	10	100	100	1	2	0.3	46.5519	**8.530144**
8	10	100	100	1	2	0.3	46.5519	**8.531728**
9	10	100	100	1	2	0.3	46.5519	**8.537607**
10	10	100	100	1	2	0.3	46.5519	**8.532742**
11	10	100	100	2	3	0.5	46.5519	**8.757501**
12	10	100	100	2	3	0.5	46.5519	**8.574803**
13	10	100	100	2	3	0.5	46.5519	**8.584994**
14	10	100	100	2	3	0.5	46.5519	**8.647263**
15	10	100	100	2	3	0.5	46.5519	**8.658588**
16	10	100	100	2	3	0.5	46.5519	**8.748089**
17	10	100	100	2	3	0.5	46.5519	**8.594535**
18	10	100	100	2	3	0.5	46.5519	**8.743204**
19	10	100	100	2	3	0.5	46.5519	**8.732453**
20	10	100	10	2	3	0.5	46.5519	**0.996702**
21	10	100	10	2	3	0.5	46.5519	**0.985875**
22	10	100	10	2	3	0.5	46.5519	**0.980589**
23	10	10	10	2	3	0.5	46.5519	**0.991487**
24	10	10	9	2	3	0.5	46.5519	**0.985393**
25	10	10	8	2	3	0.5	46.5519	**0.988249**
26	10	10	7	2	3	0.5	46.5519	**0.990082**
27	10	10	6	2	3	0.5	46.5519	**0.989426**
28	10	10	5	2	3	0.5	46.5519	**0.98293**
29	10	10	4	2	3	0.5	46.5519	**0.979736**
30	10	10	3	2	3	0.5	46.5519	**0.992027**
31	22	100	100	2	3	0.5	75.3097	20.329047
							5.813855967	**Seconds**
							20.329047	**Seconds**

Table 4. Parallel Variant for MMAS

EXPERIMENT	CITIES	ANTS	ITERATIONS	ALPHA	BETA	RHO	MINIMUN DISTANCE	lab1	lab2	lab3	lab4
1	10	100	100	1	2	0.3	46.5519	15.616404	15.656199	15.725126	**15.825714**
2	10	100	100	1	2	0.3	46.5519	15.678724	15.587746	15.675984	**15.923952**
3	10	100	100	1	2	0.3	46.5519	15.774443	15.751579	15.747124	**15.884351**
4	10	100	100	1	2	0.3	46.5519	**15.771519**	15.598393	15.7615	15.716631
5	10	100	100	1	2	0.3	46.5519	15.910909	15.829529	15.946475	**16.128292**
6	10	100	100	1	2	0.3	46.5519	15.742674	15.663364	15.72186	**15.988445**
7	10	100	100	1	2	0.3	46.5519	15.708747	15.692472	15.700473	**15.800296**
8	10	100	100	1	2	0.3	46.5519	15.767879	15.576825	15.869953	**15.948438**
9	10	100	100	1	2	0.3	46.5519	15.919635	15.658901	**15.972495**	15.800514
10	10	100	100	1	2	0.3	46.5519	15.724937	15.706604	15.862844	**15.87672**
11	10	100	100	2	3	0.5	46.5519	15.861694	15.737939	15.916608	**16.02257**
12	10	100	100	2	3	0.5	46.5519	**15.872514**	15.749501	15.821333	15.8214
13	10	100	100	2	3	0.5	46.5519	**15.975199**	15.763913	15.827095	15.844473
14	10	100	100	2	3	0.5	46.5519	15.827191	15.798001	15.831821	**16.165865**
15	10	100	100	2	3	0.5	46.5519	15.827191	15.798001	15.831821	**16.165865**
16	10	100	100	2	3	0.5	46.5519	15.891914	15.836529	16.037666	**16.1051**
17	10	100	100	2	3	0.5	46.5519	15.874714	15.852839	**16.935693**	15.865242
18	10	100	100	2	3	0.5	46.5519	15.87535	15.805074	**15.996528**	15.843767
19	10	100	100	2	3	0.5	46.5519	15.866498	15.82524	15.784151	**16.050466**
20	10	100	100	2	3	0.5	46.5519	15.850193	15.708793	15.816849	**16.069687**
21	10	100	100	2	3	0.5	46.5519	15.823103	15.77423	**16.148439**	15.962383
22	10	100	100	2	3	0.5	46.5519	15.872622	15.737472	**16.052308**	15.989298
23	10	10	10	2	3	0.5	46.5519	1.765158	**1.765994**	1.761965	1.729142
24	10	10	9	2	3	0.5	46.5519	1.737158	**1.790317**	1.788179	1.740829
25	10	10	8	2	3	0.5	46.5519	1.750755	1.71053	1.783875	**1.794154**
26	10	10	7	2	3	0.5	46.5519	1.778715	1.790239	1.770882	**1.825998**
27	10	10	6	2	3	0.5	46.5519	1.757422	1.717733	1.760698	**1.844163**
28	10	10	5	2	3	0.5	46.5519	**1.786446**	1.754178	1.772288	1.771232
29	10	10	4	2	3	0.5	46.5519	1.741479	1.724236	**1.784747**	1.783529
30	10	10	3	2	3	0.5	46.5519	**1.80604**	1.687742	1.78379	1.770516
31	22	100	100	2	3	0.5	75.3097	38.742169	38.24625	37.974074	38.057135
								3.058986 Seconds 9.6855423 Seconds			

5 Conclusions

Experiments were performed using the same parameters and processes executing sequentially and parallel form with different algorithms, ACS and MMAS. Similar results were obtained but with less processing time a parallel implementation. In the experiments where we used 1000 cities with variant ACS, time is doubled in the sequential implementation with a time of 16 minutes and in parallel only use 8 minutes. Subsequently, the variant MMAS was performed an experiment with 22 cities and the best processing was of 9.68 seconds obtained in parallel. Using a parallel implementation we can run complex optimization problems, for example, using a larger number de cities and getting results with less processing time.

References

[1] Almirón, M., Barán, B., Chaparro, E.: Ant Distributed System for Solving the Traveling Salesman Problem. In: XXV lnformatic Latinoamerican Conf.-CLEI, Paraguay, pp. 779–789 (1999)

[2] Barán, B., Sosa, R.: A New approach for AntNet routing. In: IEEE Ninth International Conference onComputer Commnunications and Networks, Las Vegas, Nevada (2000)
[3] de la Cruz, J., Mendoza, A., del Castillo, A., Paternina, C.: Comparative Analysis of heuristic Approaches Ant Q, Simulated Annealing and Tabu Search in Solving the Traveling Salesman. Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S. Ingenieria Informática, Granada, España (2003)
[4] Dorigo, M., Stutzle, T.: Ant Colony Optimization. Massachusetts Institute of Technology, MIT Press, Bradford, Cambridge (2004)
[5] Favaretto, D., Moretti, E., Pellegrini, P.: Ant colony system for variants of traveling salesman problem with time windows, Technical Report. Applied Mathematics Department of Ca' Foscari University of Venice, No. 120/2004 (2004)
[6] Cordón, O., Moya, F., Zarco, C.: A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems. Soft Computing 6(5), 308–319 (2002)
[7] Pavez, A., Acevedo, H.: An Algorithm ACS Motion Prompt and Operator 2-Opt, Departamento de Informatica. Universidad Técnica Federico Santa Maria (2002)
[8] Website of Ant Colony Optimization Algorithms official, http://www.aco-metaheuristic.org (accessed May 5, 2012)
[9] Website of interface design, http://www.matpic.com, MC. Diego O. Barragán Guerrero, Universidad Estatal de Campinas, Brasil, http://www.unicamp.br (accessed May 2012)
[10] Website of Matlab, http://www.mathworks.com (accessed May 2012)