A Preliminary Study for Multiple Ant Colony System with New Communication Strategies

ISSMAIL ELLABIB[‡] OTMAN BASIR[§]

PAUL CALAMAI[‡]

Dept. of Systems Design Engineering[‡] Dept. of Electric and Computer Engineering[§] University of Waterloo, Ontario, Canada, N2L 3G1

Abstract

In this paper we apply the concept of parallel processing to enhance the performance of the Ant Colony System algorithm. New communication strategies based on a weighting scheme are introduced under three different types of interactions. The performance of the Multiple Ant Colony System employing these strategies applied to the Traveling Salesman Problem is investigated and evaluated with respect to solution quality and computational effort. The results demonstrate that the Multiple Ant Colony System outperforms the sequential Ant Colony System. The study indicates that the weighting scheme yields a positive influence on performance, particularly in strategies that share pheromone information among all colonies.

Key-Words : Communication strategies, Parallel Ant Colony Optimization, Traveling Salesman Problem

1 Introduction

The basic idea behind ACO algorithms is to simulate the foraging behavior of a swarm of real ants (i.e., colony) using artificial ants working as cooperative agents to construct high quality solutions by using a construction procedure. This procedure manages a colony of ants that concurrently and asynchronously constructs problem solutions by moving through neighbor nodes of the problem's construction graph [1]. This mechanism can make an ACO algorithm well suited for parallelization. Bullnheimer et al. [2] introduced two parallel implementations of the Ant System (AS) algorithm, namely, the Synchronous Parallel Implementation (SPI) and the Partially Asynchronous Parallel Implementation (PAPI). SPI is based on a master-slave paradigm in which every ant finds a solution in the slave and sends the result to the master. The master updates the pheromone information, when the solutions are available from all slaves, and sends the updated information back to all slaves. This implementation parallelizes the construction phase, but it has the disadvantage that all ants have to wait for each other at every iteration due to a communication overhead between the master and the slaves. PAPI is based on the coarse-grained model in which information is exchanged among colonies every fixed number of iterations. The experiential results obtained by the authors indicated that PAPI performs better than SPI in terms of running time and speedup. Talbi et al. [3] introduced another masterslave paradigm to parallelize the AS algorithm in which a local search method based on Tabu Search (TS) is introduced in each slave to improve the solution constructed by the ant. Their results demonstrated the complementary gains brought by incorporating TS within the AS algorithm in the parallel implementation. Middendorf et al. [4] introduced a parallel implementation, based on the coarse-grained model, of the AS algorithm. Their multiple colony approach employs best-exchange strategies in which best solution information is exchanged between colonies under two different communication topologies.

They studied the performance of employing these strategies and the results indicated that the multiple AS algorithm employing some strategies outperforms the single AS algorithm in certain instances. Chu et al.[5] introduced a parallel implementation, based on the coarsegrained model, of the Ant Colony System algorithm and used the same idea of the best solution exchange but under several communication topologies. Experiments indicated that the Parallel Ant Colony System algorithm outperforms both the AS and the ACS algorithms.

In general, ACO is inherently a distributed methodology, so it is particularly suited to parallelization [1]. Although a number of parallel versions of ACO have been implemented and tested in limited settings, it is still an open question as to how an efficient parallel version of an ACO algorithm should be implemented, and how much improvement can be obtained over the sequential ACO algorithm. As well, since performance can be enhanced not only by the parallel implementation but also by the efficiency of the communication strategies that are incorporated in the parallel implementation, these strategies need to be studied.

This paper is organized as follows. In Section 2 we describe the Ant Colony System. In Section 3 the concept of parallel processing is applied to the Ant Colony System, and communication strategies based on a weighting scheme are introduced. Experimental results from the Multiple Ant Colony System are presented in Section 4 along with a comparative performance analysis involving other existing approaches. Finally, Section 5 provides some concluding remarks.

2 Ant Colony System

The Ant Colony System (ACS) algorithm features major changes in the transition and pheromone update rules of the AS algorithm [6]. A new transition rule is introduced that favors either exploitation or exploration according to the following transition rule:

$$P_{ij}^{k} = \begin{cases} 1 & if\{j = j^{*} & \& q \leq q_{0}\} \\ 0 & if\{j \neq j^{*} & \& q \leq q_{0}\} \\ \frac{[\tau_{ij}][\eta_{ij}]^{\beta}}{\sum_{u \in J_{i}^{k}} [\tau_{iu}][\eta_{iu}]^{\beta}} & otherwise \end{cases}$$
(1)

where $j^* = \arg \max_{u \in J_i^k} ([\tau_{iu}][\eta_{iu}]^\beta)$, τ_{ij} is the amount of pheromone on the edge joining nodes *i* and *j*, η_{ij} is

the heuristic information for the ant visibility measure (e.g., defined as the reciprocal of the distance between node *i* and node *j* for TSP), and β is a control parameter that represents the relative importance of the ant visibility value versus the amount of pheromone on the edge joining nodes *i* and *j*. *q* is a generated random number in the range [0,1], and q_0 is a given threshold parameter. Thus, when *q* is less than or equal to q_0 the ant employs exploitation to select node j^* as the next node in its tour, whereas if *q* exceeds q_0 the ant uses probabilistic exploration to select the next node in its tour. From node *i*, the next node *j* in the route is selected by ant *k* among the unvisited nodes J_i^k .

The pheromone is updated in two different ways:

• *Local updating*: As the ant moves between nodes *i* and *j*, it updates the amount of pheromone on the visited edge using the following formula

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \tag{2}$$

where τ_0 , the initial amount of pheromone^a, is calculated as $\tau_0 = (nC_i)^{-1}$, n is the problem size (i.e., the number of nodes) and C_i is the cost of the initial tour produced by a construction heuristic such as the Nearest Neighbor (NN) heuristic, and ρ , the evaporation rate, is a parameter in the range [0,1] that regulates the reduction of pheromone on the edges. The effect of local updating is that each time an ant uses an edge(i,j) its pheromone trail τ_{ij} is reduced, so that edges becomes less desirable for the ants at the next iterations. This permits an increase in the exploration of edges that have not been visited yet. In fact, local updating has helped to avoid poor stagnation situations^b.

• *Global updating*: When all ants have generated their tours, the edges belonging to the best tour are updated using the following formula:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho(1/C_b)$$
 (3)

^aAt the beginning of the search a small amount of pheromone is assigned to all the edges.

^bStagnation occurs when the algorithm reaches its equilibrium state (i.e., a single path is chosen by all ants).

where C_b is the cost of the best tour found since the start of the algorithm. It is important to note that global updating adjusts only the pheromone on the edges belonging to the best tour causing ants in future iterations to search in the vicinity of this best tour.

3 Multiple Ant Colony System



Figure 1: Colony-level interaction framework.

We believe that the parallel running of a group of colonies can be enhanced by organizing those colonies in such a way that the colonies can share their information efficiently. This information can be utilized by colonies via an exchange module that defines the interaction between the group of connected colonies. The interaction between the colonies relies on the colonies adopting an efficient communication architecture that facilitates cooperation between them, and a communication strategy that defines the rules for collaboration among them. Accordingly, we define an exchange module as shown in Figure 1 in which every colony handles the mechanism of cooperation.

In the proposed approach, a group of identical colonies search in parallel and communicate with each other via the exchange module. The ants in each colony are divided equally into several groups. Each colony is associated with a complete ACS construction procedure introduced in [6]. Figure 2 illustrates the complete pseudo-code description of the Multiple Ant Colony System (M-ACS) algorithm. In this algorithm, colony h provides its search information to other colonies and receives search information from other colonies at every fixed number of iterations (i.e., exchange interval) via the exchange module procedure. In this procedure, the interaction occurs under different topologies and according to one of the communication strategies described below.

- 1: apply the NN heuristic to generate an initial feasible solution for all colonies
- 2: while termination condition is not met do
- 3: for each colony $h \in S$ in parallel do
- 4: set an initial value of pheromone on every edge
- 5: iteration number=1
- 6: **for** k = 1 to m ants **do**
- 7: apply the construction procedure of ACS
- 8: end for
- 9: **if** iteration number *mod* exchange interval=0 **then**
- 10: improve the solutions by local search procedure {option}
- 11: call the exchange module procedure
- 12: **end if**
- 13: increment iteration number
- 14: **end for**
- 15: end while

16: display the best solution found so far by all colonies17: stop

Figure 2: A pseudo-code description of the M-ACS algorithm.

In our parallel implementation of the algorithm, each colony consists of a swarm of 10 ants managed by one machine. Machine 0 is responsible for initialization, spawning, and collection and display of the results, while all machines (including machine 0) are responsible for constructing solutions to the problem. A Beowulf cluster is used for this implementation. The Beowulf cluster consists of a collection of PC machines interconnected by a Local Area Network (LAN) running the Red Hat Linux operating system. We employed the Message Passing Interface (MPI) software to allow the Beowulf cluster machines to interact. We implemented, in the C++ language, the M-ACS algorithm with the option of employing one of the communication strategies described below. The parallel implementation combines a group of 8 identical machines of the cluster equipped with 700MHz Pentium III processors and 128MB of RAM each.

3.1 Communication Strategies

The search for high quality solutions can be improved, in terms of performance, by employing a communication strategy that can propagate current high quality solution information to the colonies and use that information in the pheromone representation.

In the proposed strategies, we adjust the pheromone matrix of each colony through different colony-level interactions and according to the solution information so as to reinforce search in the vicinity of high quality solutions. The behavior of ants in one colony will be influenced by the solution information received from other colonies, where pheromone is added to the colony edges that belong to the best solutions of the group of colonies.

A weighting scheme similar to the one presented in [7] is applied in the proposed multiple colony approach to assess the quality of the best solutions constructed by several colonies. The pheromone trails on the edges of the best solutions are updated adaptively in response to determined weights, and an extra amount of pheromone is deposited on the edges of these solutions accordingly.

To identify whether the colonies are converging toward one solution or scattered in the search space we calculate the difference between the current overall average cost of the best solutions produced by the selected colonies and the cost of the best solution found so far. The cost of the best solution found so far is given by $C_b = min_{h \in S} \{C_b^h\}$. The overall average cost of the best solutions is given by $C_{AVG} = \frac{1}{|S|} \sum_{h \in S} C_b^h$, where C_b^h is the cost of the best solution found by colony h, for $h \in S$ where $S \subset \{1, ..., M\}$, and M is the total number of colonies. We note that the difference between the overall average cost C_{AVG} and the best cost C_b is likely to be less when the selected colonies approach the best solution than it will be when these colonies are scattered in the search space. We therefore use the difference $(C_{AVG} - C_b)$ as a yardstick for detecting the convergence of the selected colonies. The colony $h \in S$ that has the best solution C_b^h that is less than the overall average C_{AVG} is assigned a weight $w^h \in (0, 1]$, otherwise it is assigned a weight equal to zero as given by

$$w^{h} = \begin{cases} \frac{C_{AVG} - C_{b}^{h}}{C_{AVG} - C_{b}} & if\{h \in S, (C_{AVG} - C_{b}^{h}) > 0\}\\ 0 & otherwise \end{cases}$$
(4)

which depends not only on the measure of convergence previously discussed but also on how close the cost of the best solution for the colony, C_b^h , is to the cost of the best solution found so far, C_b , in such a way that the closer C_b^h is to C_b the closer w^h is to 1. These weights are used to define a colony-level interaction pheromone update formula intended to achieve a trade-off between exploration and exploitation. The following update formula is applied to edges of the best tour $T_b^h \forall h \in S$ as given by

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho(w^h/C_b^h) \ if\{edge(i, j) \in T_b^h\}$$
 (5)

In fact, this formula has the effect of increasing the quantity of pheromone on edges associated with some solutions according to the quality of these solutions and the current convergence state of the colonies.



Figure 3: Interaction of colonies in a star topology.



Figure 4: Interaction of colonies in a hypercube topology.



Figure 5: Interaction of colonies in a directed ring topology.

The exchange module exchanges the best solutions between a group of colonies (specified by the selected topology), weighs those best solutions using the weighting scheme, and then applies the colony-level interaction pheromone update formula. Accordingly, we define the communication strategies as follows:

• Strategy-1: the interaction is applied among *M* colonies organized as in Figure 3.

- Strategy-2: the interaction is applied among |S| colonies (i.e., S ⊂ {1, ..., M}) organized as in Figure 4.
- Strategy-3: the interaction is applied between two consecutive colonies organized as in Figure 5.

3.2 Influence of the Communication Strategies

The search behavior of ACO algorithms can be visualized and assessed through the distribution of pheromone trail values in the pheromone matrix. We define the en*tropy* of node *i* at iteration *t*, $\mathcal{H}_i(t)$, as given by $\mathcal{H}_i(t) =$ $-\sum_{i \in N} \tau_{ii}(t) \log \tau_{ii}(t)$ and interpret this as a measure of the (potential) diversity in the search from this node at that iteration. The team consensus methodology, introduced in [8], is applied to aggregate the node entropies to arrive at the cumulative entropy of each colony. In this section, we study the influence of the exchange strategies on the search behavior of the M-ACS algorithm for solving the TSP test problem eil101 [9]. The cumulative entropies of the M colonies are reported and averaged at every iteration t = 1, ..., 500 for M-ACS without employing any strategy (No-Strategy^c) as well as employing the three communication strategies (Strategy-1, Strategy-2, Strategy-3).

In order to determine an appropriate exchange interval I (the number of iterations between exchanges), different values are considered for these strategies. The appropriate value of I is selected according to the diversity level prior to the exchange (i.e., when the search is diversified), and the fluctuation range—the difference in the entropy prior to the exchange and immediately after the exchange (i.e., when the search is intensified). The diversity level and the fluctuation range for I = 30 are generally greater than for other exchange intervals. For this reason the exchange interval I = 30 was used for the computational tests.

For the appropriate value of I, Figure 6 illustrates the average cumulative entropies of No-Strategy and the three communication strategies. As illustrated, Strategy-1 exhibits the largest swing (largest fluctuation) in cumulative entropy about every exchange step. This strategy also intensifies the search more than the other strategies (i.e., has lower entropy at the exchange step) and contrasts with



Figure 6: Average cumulative entropy of the communication strategies with I = 30, TSP test problem: *eil101*.

Strategy-3, which diversifies the search the most (i.e., has the highest entropy prior to the exchange step). This indicates that the strategies produce searches that are different in their ability to intensify and diversify the search at every exchange step.

4 Experimental Results

A performance study was carried out to evaluate the effectiveness of the communication strategies and compare the performance of the M-ACS with the performance of the Ant Colony System (ACS) and Parallel Ant Colony System (PACS) of Chu et al. [5] using some TSP problem instances [9]. For all approaches, the parameter settings, as proposed in [6], were set to $\beta = 2$, $\rho = 0.1$, $q_0 = 0.9$, and m = 10 ants. To ensure a fair comparison, we set the number of runs and the number of iterations consistent with the ACS and the PACS approaches. The performance is evaluated on the basis of the best solutions obtained in several runs on each instance. The number of iterations for problems st70 and eil101 was set to 1,000 iterations, and for problem *tsp225* was set to 2,000. Table 1 summarizes the performance in terms of the Mean (Mn) and the Standard Deviation (SD) of the best solutions obtained over 10 runs. The results illustrate the potential of applying the multiple version over the sequential version of ACS with some variations in the performance of the multiple version employing different strategies with Strategy-

^cM colonies work in parallel without interaction

	Stra-	Stra-	Stra-
PACS	tegy-1	tegy-2	tegy-3
3 678.8	675.6	676.1	676.7
3.3	1.1	1.0	1.4
2 646.3	638.1	639.6	642.1
4.3	3.6	5.3	3.14
4.4 3887.0	3936.7	3942.4	3971.8
12.9	14.6	17.4	39.00
	PACS 3 678.8 3.3 2 2 646.3 4.3 3887.0 12.9 12.9	Sura- PACS tegy-1 3 678.8 675.6 3.3 1.1 2 646.3 638.1 4.3 3.6 4.4 3887.0 3936.7 12.9 14.6	Sua- Sua- Sua- S PACS tegy-1 tegy-2 3 678.8 675.6 676.1 3.3 1.1 1.0 2 646.3 638.1 639.6 4.3 3.6 5.3 4.4 3887.0 3936.7 3942.4 12.9 14.6 17.4

Table 1: Performance comparison between ACS, PACS, and M-ACS employing Strategy-1, Strategy-2, and Strategy-3.

1 exhibiting better performance than any other strategy. Furthermore, the SD of Strategy-1 is quite small compared to the others, which shows the consistency of M-ACS employing Strategy-1. Note that the results of PACS reported in this table were obtained from the best results of employing several communication strategies. The authors also report 3887.0 for the Mn value of tsp225, but the best known solution is 3916 [9]. In general, the results obtained by M-ACS employing Strategy-1 demonstrate the effectiveness and the consistency of Strategy-1 compared with the other tested strategies for the three instances.

5 Discussion

The important issue in the ACO algorithms is to find useful mechanisms for sharing information to improve search behavior. In this paper, the single colony approach is extended to a multiple colony approach. Communication strategies are introduced under three different types of interactions. A weighting scheme is applied in these strategies for adapting the amount of pheromone based on the quality of solutions found by several colonies, considering the state of convergence. A performance study is performed on TSP instances to confirm the effectiveness of these strategies. The results indicated the domination of the Multiple Ant Colony System over the sequential Ant Colony System. In particular, a significant improvement can be accomplished by employing the strategy executed under the star topology, in which the search information is shared among all colonies. Based on this evidence we postulate that the proposed multiple colony approach is a promising approach for solving the TSP and possibly other combinational optimization problems, however, further investigation to determine the optimal interaction

scheme among colonies is still needed. Of particular importance is the development of an interaction scheme in which the colonies can interact asynchronously.

References

- [1] Dorigo M. and Stützle T. *Ant Colony Optimization*. Massachusetts Institute of Technology, 2004.
- [2] Bullnheimer B., Kotsis G., and Strauss C. Parallelization Strategies for Ant System. In DeLeone R., Murli A., Pardalos P., and Toraldo G. (eds.), *High Performance Algorithms and Software in Nonlinear Optimization*, vol. 24 of *Applied Optimization*. Kluwer, Dordrecht, 1998, pp. 87–100.
- [3] Talbi E., Roux E., Fonlupt C., and Robillard D. Parallel Ant Colonies for Combinatorial Optimization Problems. In et al. J.R. (ed.), *Parallel and Distributed Processing*, 11 IPPS/SPDP'99 Workshops, vol. 1586 of LNCS. Berlin: Springer, 1999.
- [4] Middendorf M., Reischle F., and Schmeck H. Multi Colony Ant Algorithms. *Journal of Heuristics*, vol. 8(3), 2002, pp. 305–320. Special issue on Parallel Metaheuristics.
- [5] Chu S.C., Roddick J.F., and Pan J.S. Ant Colony System with Communication Strategies. *Information Sciences*, vol. 167, 2004, pp. 63–76.
- [6] Dorigo M. and Gambardella L. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Trans. on Evol. Comp.*, vol. 1, 1997, pp. 53–66.
- [7] Ellabib I., Basir O., and Calamai P. A New Ant Colony System updating strategy for the Vehicle Routing Problems with Time Windows. In the Proceedings of the Fifth Metaheuristics International Conference (MIC2003). Kyoto, Japan, 2003.
- [8] Basir O. and Shen H. New Approach for Aggregating Multi-Sensory Data. *Journal of Robotic Systems*, vol. 10(8), 1993, pp. 1075–1093.
- [9] Reinelt G. TSPLIB library. Http:// www.iwr.uniheidelberg.de/ groups/ comopt/ software/ TSPLIB95/.