Parallel Ant Colony Systems

Shu-Chuan Chu[1], John F. Roddick[1], Jeng-Shyang Pan[2,3], and Che-Jen Su[2]

[1] School of Informatics and Engineering,
Flinders University of South Australia,
GPO Box 2100, Adelaide 5001, South Australia.
{shuchuan.chu, roddick}@infoeng.flinders.edu.au

[2] Department of Electronic Engineering,
Kaohsiung University of Applied Sciences
Kaohsiung, Taiwan
jspan@cc.kuas.edu.tw

[3] Department of Automatic Test and Control,
Harbine Institute of Technology
Harbine, China

Abstract. In this paper a Parallel Ant Colony System ($PACS$) is developed. Three communication methods for updating the pheromone level between groups in $PACS$ are proposed and work on the traveling salesman problem using our system is presented. Experimental results based on three well-known traveling salesman data sets demonstrate the proposed $PACS$ is superior to the existing Ant Colony System (ACS) and Ant System (AS) with similar or better running times.

1 Introduction

Swarm intelligence research originates from work into the simulation of the emergence of collective intelligent behavior of real ants. Ants are able to find good solutions to the shortest path between the nest and a food source by laying down, on their way back from the food source, a trail of an attracting substance – a *pheromone*. Communication is based on the pheromone level – the shortest path being considered that with the greatest density of pheromone and with ants tending to follow the path with more pheromone. Dorigo *et al.* were the first to apply this idea to the traveling salesman problem [1,2]. The initial algorithm was referred to as the *Ant System* (AS) algorithm and a later enhanced method was also developed and referred to as the *Ant Colony System* (ACS) algorithm [3]. The Ant System and Ant Colony System algorithms have been applied successfully in many applications such as the quadratic assignment problem [4], data mining [5] and space-planning [6].

Parallelization strategies for AS [7] and ACS [8] have been investigated, however, these studies are based on simply applying AS or ACS on the multiprocessor, ie. the parallelization strategies simply share the computation load over several processors. No experiments demonstrate the sum of the computation time for all processors can be reduced compared with the single processor works on the AS or ACS.

N. Zhong et al. (Eds.): ISMIS 2003, LNAI 2871, pp. 279–284, 2003.

In this paper, we apply the concept of parallel processing to the Ant Colony System (ACS) and a *Parallel Ant Colony System* ($PACS$) is proposed. The purpose of the $PACS$ is not just to reduce the computation time. Rather a parallel formulation is developed which gives not only reduces the elapsed and the computation time but also obtains a better solution. The artificial ants are firstly generated and separated into several groups. The Ant Colony System is then applied to each group and communication between groups is applied according to some fixed cycles. The basic idea of the communication is to update the pheromone level for each route according to the best route found by neighbouring groups or, in some cases, all groups. Three communication methods are proposed for $PACS$. Experimental results based on the traveling salesman problem confirm the efficiency and effectiveness of the proposed $PACS$.

2 Parallel Ant Colony System

A parallel computer consists of a large number of processing elements which can be dedicated to solving a single problem at a time. Pipeline processing and data parallelism are two popular parallel processing methods. Data parallelism has been applied to genetic algorithms by dividing the population into several groups and running the same algorithm over each group using different processor [9]. The resulting parallel genetic algorithm has been successfully applied to noise reduction of vector quantization based communication [10]. In this paper, we apply the idea of data parallelism to Ant Colony Systems (ACS) in order to reduce running time and obtain a better solution. The Parallel Ant Colony System ($PACS$) is described as follows:

Step 1: Initialization – Generate N_j artificial ants for the jth group, $j = 0, 1 \ldots G-1$. N_j and G are the number of artificial ants for the jth group and the number of groups, respectively. Randomly select an initial *city* for each ant. The initial pheromone level between any two cities is set to be a small positive constant τ_0. Set the cycle counter to be 0.

Step 2: Movement – Calculate the next visited city s for the ith ant in the jth group according to

$$s = \begin{cases} arg\ max_{u \in J_{i,j}(r)} [\tau_j(r,u)] \cdot [\eta(r,u)]^\beta & , \quad if\ q \le q_0 \quad (exploitation) \\ P_{i,j}(r,s) & , \quad otherwise \quad (biased\ exploration) \end{cases}$$

$$P_{i,j}(r,s) = \begin{cases} \frac{[\tau_j(r,s)] \cdot [\eta(r,s)]^\beta}{\sum_{u \in J_k(r)} [\tau_j(r,u)] \cdot [\eta(r,u)]^\beta} & , \quad if\ s \in J_{i,j}(r) \\ 0 & , \quad otherwise \end{cases}$$

where $P_{i,j}(r,s)$ is the transition probability from city r to city s for the ith ant in the jth group. $\tau_j(r,s)$ is the pheromone level between city r to city s in the jth group. $\eta(r,s) = \frac{1}{\delta(r,s)}$ the inverse of the distance $\delta(r,s)$

between city r and city s. $J_{i,j}(r)$ is the set of cities that remain to be visited by the ith ant in the jth group and β is a parameter which determines the relative importance of pheromone level versus distance. q is a random number between 0 and 1 and q_0 is a constant between 0 and 1.

Step 3: Local Pheromone Level Updating Rule – Update the pheromone level between cities for each group as

$$\tau_j(r,s) \longleftarrow (1-\rho) \cdot \tau_j(r,s) + \rho \cdot \Delta\tau(r,s)$$

$$\Delta\tau(r,s) = \tau_0 = (n * L_{nn})^{-1}$$

where $\tau_j(r,s)$ is the pheromone level between cities r and s for the ants in the jth group, L_{nn} is an approximate distance of the route between all cities using the *Nearest Neighbour Heuristic*, n is the number of cities and $0 < \rho < 1$ is a pheromone decay parameter. Continue Steps 2 and 3 until each ant in each group completes the route.

Step 4: Evaluation – Calculate the total length of the route for each ant in each group.

Step 5: Global Pheromone Level Updating Rule – Update the pheromone level between cities for each group as

$$\tau_j(r,s) \longleftarrow (1-\alpha) \cdot \tau_j(r,s) + \alpha \cdot \Delta\tau_j(r,s)$$

$$\Delta\tau_j(r,s) = \begin{cases} (L_j)^{-1} & , \; if \ (r,s) \in best\ route\ of\ jth\ group \\ 0 & , \; otherwise \end{cases}$$

where L_j is the shortest length for the ants in the jth group and α is a pheromone decay parameter.

Step 6: Updating from Communication – Three communication methods are proposed as follows:

- **Method 1:** Update the pheromone level between cities for each group for every R_1 cycles as

$$\tau_j(r,s) \longleftarrow \tau_j(r,s) + \lambda \cdot \Delta\tau_{best}(r,s)$$

$$\Delta\tau_{best}(r,s) = \begin{cases} (L_{gb})^{-1} & , \; if \ (r,s) \in best\ route\ of\ all\ groups \\ 0 & , \; otherwise \end{cases}$$

where λ is a pheromone decay parameter and L_{gb} is the length of the best route of all groups, i.e., $L_{gb} < L_j$, $j = 0, 1 \ldots G-1$.

- **Method 2:** Update the pheromone level between cities for each group for every R_2 cycles as

$$\tau_j(r,s) \longleftarrow \tau_j(r,s) + \lambda \cdot \Delta\tau_{ng}(r,s)$$

$$\Delta\tau_{ng}(r,s) = \begin{cases} (L_{ng})^{-1} & , \ if\ (r,s) \in best\ route\ of\ neighbour\ group \\ 0 & , \ otherwise \end{cases}$$

where neighbour is defined as being the group whose binary representation of the group number j differs by the least significant bit. λ is a pheromone decay parameter and L_{ng} is the length of the shortest route in the neighbour group.

- **Method 3:** Update the pheromone level between cities for each group using both Method 1 and Method 2.

Step 7: Termination – Increment the cycle counter. Move the ants to the originally selected cities and continue Steps 2 to 6 until the stagnation or a present maximum number of cycles has reached, where a stagnation indicated by all ants taking the same route.

3 Experimental Results

Experiments were carried out to test the performance of the Ant System (AS), Ant Colony System (ACS) and Parallel Ant Colony System ($PACS$) for the traveling salesman problem. Three generally available and typical data sets, EIL101, ST70 and TSP225 were used as the test material[1]. EIL101, ST70 and TSP225 are data sets with 101, 70 and 225 cities, respectively. The number of ants for AS, ACS and $PACS$ was set to be 80, with the ants divided into 4 groups of 20 for $PACS$. All results shown are averaged over 5 runs. The parameters were set to the following values: $\beta = 2$, $q_0 = 0.9$, $\alpha = \rho = \lambda = 0.1$. The number of cycles between updates of the pheromone level from communication for methods 1 and 2 in $PACS$ were set to 80 and 30, respectively.

The data sets for the first and second experiment were EIL101 and ST70. We recorded the shortest length of the route for a fixed running time of 300 secs. The TSP225 data set was used for the third experiment and the shortest length of the route for an execution time of 600 secs was recorded. $PACS$ with method 1, method 2 and method 3 for communication are referred to as $PACS1$, $PACS2$ and $PACS3$, respectively. The experimental results of the data sets EIL101 for one seed are shown in Figure 1. The experimental results shown in Table 1 are the average shortest length over 5 runs for all data sets.

[1] available from *http://www.iwr.uniheidelberg.de/groups/comopt/software/*

The average improvement on three data sets for the proposed $PACS1$, $PACS2$ and $PACS3$ compared with ACS was 4.17%, 4.11% and 4.35%, respectively. Compared with AS, the $PACS1$, $PACS2$ and $PACS3$ shows improvements of 9.32%, 9.27% and 9.49%, respectively.

Table 1. Performance Comparison of the shortest length for Three Datasets

Data sets	AS	ACS	$PACS1$	$PACS2$	$PACS3$
EIL101	718.5917	670.8689	646.4660	648.6656	645.9640
TSP225	4536.8308	4199.8483	3939.4237	3934.4802	3916.0839
ST70	711.4132	696.1061	677.4363	677.1765	678.5531

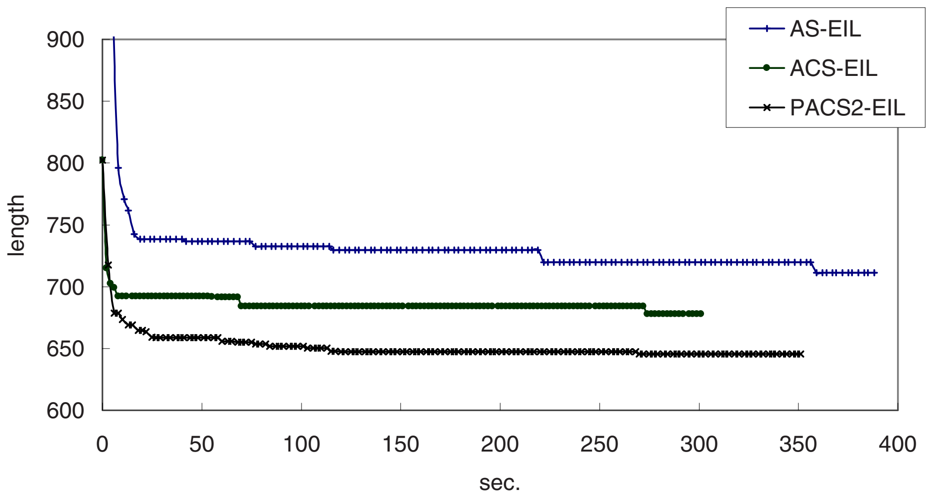

Fig. 1. Performance Comparison of AS, ACS, $PACS2$ for EIL101.tsp data sets

4 Conclusions

The main contribution of this paper is to propose a parallel formulation for the Ant Colony System (ACS). Three communication methods between groups which can be used to update the pheromone levels are presented. From our experiments, the proposed Parallel Ant Colony System ($PACS$) outperforms both ACS and AS based on three available traveling salesman data sets. We will continue to work on new communication strategy for updating the pheromone level in $PACS$ and we are interested in applying the $PACS$ to data clustering in future.

References

1. A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *First Europ. Conference Artificial Life* (F. Varela and P. Bourgine, eds.), pp. 134–142, 1991.
2. M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
3. J. M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
4. V. Maniezzo and A. Colorni, "The ant system applied to the quadratic assignment problem," *IEEE Trans. on Knowledge and Data Engineering*, vol. 11, no. 5, pp. 769–778, 1999.
5. R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 4, pp. 321–332, 2002.
6. J. A. Bland, "Space-planning by ant colony optimization," *International Journal of Computer Applications in Technology*, vol. 12, no. 6, pp. 320–328, 1999.
7. B. Bullnheimer, G. Kotsis, and C. Strauss, "Parallelization strategies for the ant system," Technical Report POM 9/97, Institute of Applied Computer Science, University of Vienna, Austria, 1997.
8. T. Stutzle, "Parallelization strategies for ant colony optimization," in *Fifth International Conference on Parallel Problem Soving for Nature*, pp. 722–731, LNCS 1498, Springer-Verlag Berlin Heidelberg, 1998.
9. J. P. Cohoon, S. U. Hegde, W. N. Martine, and D. Richards, "Punctuated equilibria: a parallel genetic algorithm," in *Second International Conference on Genetic Algorithms*, pp. 148–154, 1987.
10. J. S. Pan, F. R. McInnes, and M. A. Jack, "Application of parallel genetic algorithm and property of multiple global optima to vq codevector index assignment for noisy channels," *Electronics Letters*, vol. 32, no. 4, pp. 296–297, 1996.