Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences 167 (2004) 63–76
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Ant colony system with communication strategies

Shu-Chuan Chu [a,b], John F. Roddick [a], Jeng-Shyang Pan [c,*]

[a] School of Informatics and Engineering, Flinders University of South Australia, P.O. Box 2100, Adelaide 5001, Australia
[b] Department of Industrial Engineering and Management, Kaohsiung University of Applied Sciences, 415 Chien-Kung Road, Kaohsiung, Taiwan ROC
[c] Department of Electronic Engineering, Kaohsiung University of Applied Sciences, 415 Chien-Kung Road, Kaohsiung, Taiwan ROC

Received 6 August 2003; received in revised form 24 October 2003; accepted 24 October 2003

Abstract

In this paper an ant colony system (ACS) with communication strategies is developed. The artificial ants are partitioned into several groups. Seven communication methods for updating the pheromone level between groups in ACS 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 ACS with communication strategies are superior to the existing ant colony system (ACS) and ant system (AS) with similar or better running times.

Keywords: Ant colony system (ACS); Communication strategies; Ant system (AS)

1. Introduction

Swarm intelligence research originates from work into the simulation of the emergence of collective intelligent behaviors of real ants. Ants are able to find

* Corresponding author. Tel.: +886-7-381-4526; fax: +886-7-389-9382.
E-mail addresses: shuchuan.chu@infoeng.flinders.edu.au (S.-C. Chu), roddick@infoeng.flinders.edu.au (J.F. Roddick), jspan@cc.kuas.edu.tw (J.-S. Pan).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2003.10.013

good solutions to the shortest path problems 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*. Based on the pheromone level communication, the shortest path is considered that with the greatest density of pheromone and the ants will tend to follow the path with more pheromone. Dorigo and his colleagues were the first to apply this idea to the traveling salesman problem [1,2]. This algorithm is referred to as *ant system* (AS) algorithm. A more promising method was also developed and referred to as the algorithm of *ant colony system* (ACS) [3]. The ant system and ant colony system have been applied successfully in many applications such as the quadratic assignment problem [4], data mining [5], space-planning [6], job-shop scheduling and graph coloring [7].

Parallelization strategies for AS [8] and ACS [9] have been investigated, however, these studies are based on simply applying AS or ACS on the multiprocessor, i.e. 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.

In this paper, we apply the concept of parallel processing to the ant colony system (ACS) and a *parallel ant colony system* (PACS) idea is proposed. ACS with communication strategies, which we termed PACS, will be used to run the experiments presented in the following of this paper. 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. Seven communication strategies are proposed for PACS. Experimental results based on the traveling salesman problem confirm the efficiency and effectiveness of the proposed PACS.

2. Ant system and ant colony system

Inspired by the food-seeking behavior of real ants, the ant system [1,2] is a cooperative population-based search algorithm. As each ant construct a route from nest to food by stochastically following the quantities of pheromone level, the intensity of laying pheromone will bias the path-choosing decision-make of subsequent ants.

The operation of ant system can be illustrated by the classical *traveling salesman problem*. A traveling salesman problem is seeking for a round route covering all cities with minimal total distance. Suppose there are n cities and m ants. The entire algorithm is started with initial pheromone intensity set to τ_0 on all edges. In every subsequent ant system cycle, or called episode, each ant begins its tour from a randomly selected starting city and is required to visit every city once and only once. The experience gained in this phase is then used to update the pheromone intensity on all edges.

The algorithm of the ant system for the traveling salesman problem (TSP) is depicted as follows [2,3]:

Step 1: Randomly select the initial city for each ant. The initial pheromone level between any two cities is set to be a small positive constant. Set the cycle counter to be 0.

Step 2: Calculate the transition probability from city r to city s for the kth ant as

$$P_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}{\sum_{u \in J_k(r)} [\tau(r,u)] \cdot [\eta(r,u)]^{\beta}} & \text{if } s \in J_k(r), \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where r is the current city, s is the next city, $\tau(r,s)$ is the pheromone level between city r and city s, $\eta(r,s) = 1/\delta(r,s)$ the inverse of the distance $\delta(r,s)$ between city r and city s, $J_k(r)$ is the set of cities that remain to be visited by the kth ant positioned on city r, and β is a parameter which determines the relative importance of pheromone level versus distance. Select the next visited city s for the kth ant with the probability $P_k(r,s)$. Repeat Step 2 for each ant until the ants have toured all cities.

Step 3: Update the pheromone level between cities as

$$\tau(r,s) \leftarrow (1-\alpha) \cdot \tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s), \tag{2}$$

$$\Delta\tau_k(r,s) = \begin{cases} \dfrac{1}{L_k} & \text{if } (r,s) \in \text{route done by ant } k, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

$\Delta\tau_k(r,s)$ is the pheromone level laid down between cities r and s by the kth ant, L_k is the length of the route visited by the kth ant, m is the number of ants and $0 < \alpha < 1$ is a pheromone decay parameter.

Step 4: Increment cycle counter. Move the ants to the originally selected cities and continue Steps 2–4 until the behavior stagnates or the maximum number of cycles has reached, where a stagnation is indicated when all ants take the same route.

From Eq. (1) it is clear ant system (AS) needs a high level of computation to find the next visited city for each ant. In order to improve the search efficiency, the ant colony system (ACS) was proposed [3]. ACS is based on AS but updates the pheromone level before moving to the next city (local updating rule) and updating the pheromone level for the shortest route only after completing the route for each ant (global updating rule) as

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s), \tag{4}$$

$$\Delta\tau(r,s) = \begin{cases} (L_{\mathrm{gb}})^{-1} & \text{if } (r,s) \in \text{global best route}, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where L_{gb} is the length of the shortest route and α is a pheromone decay parameter.

3. 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. The function of the pipeline processing is to separate the problem into a cascade of tasks where each task is executed by an individual processor, while data parallelism involves distributing the data to be processed amongst all processors which then executes the same procedure on each subset of the data. Data parallelism has been applied to genetic algorithm by dividing the population into several groups and running the same algorithm over each group using different processor [10], and the parallel genetic algorithm has been successfully applied to noise reduction of vector quantization based communication [11]. In this paper, we apply the idea of data parallelism to ant colony system (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

$$s = \arg\max_{u \in J_{i,j}(r)} [\tau_j(r,u)]\cdot[\eta(r,u)]^{\beta} \quad \text{if } q \leqslant q_0 \text{ (exploitation)},$$

visit city s with $P_{i,j}(r,s)$ if $q > q_0$ (biased exploration),

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

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. $\eta(r,s) = 1/\delta(r,s)$ 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

$$\tau_j(r,s) \leftarrow (1-\rho) \cdot \tau_j(r,s) + \rho \cdot \Delta\tau(r,s),$$
$$\Delta\tau(r,s) = \tau_0 = (n * L_{\mathrm{nn}})^{-1},$$

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

$$\tau_j(r,s) \leftarrow (1-\alpha) \cdot \tau_j(r,s) + \alpha \cdot \Delta\tau_j(r,s),$$
$$\Delta\tau_j(r,s) = \begin{cases} (L_j)^{-1} & \text{if } (r,s) \in \text{best route of } j\text{th group}, \\ 0 & \text{otherwise,} \end{cases}$$

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.* Seven communication strategies are proposed as follows:

- *Strategy 1*: As shown in Fig. 1, update the pheromone level between cities for each group for every R_1 cycles as

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

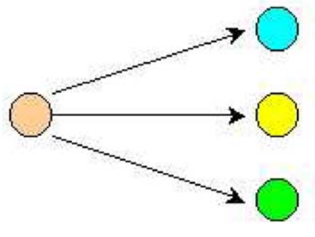

Fig. 1. Update the pheromone level according to the best route of all groups.

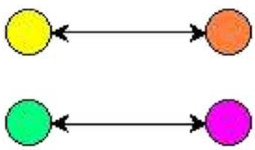

Fig. 2. Update the pheromone level between each pair of groups.

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

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

- *Strategy 2*: As shown in Fig. 2, update the pheromone level between cities for each group for every R_2 cycles as

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

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

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.

- *Strategy 3*: As shown in Fig. 3, update the pheromone between cities for each group for every R_3 cycles as

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

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

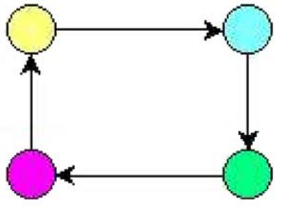

Fig. 3. Update the pheromone level according to the ring structure.

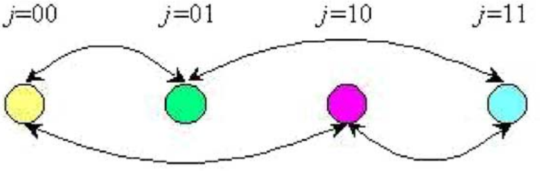

Fig. 4. Update the pheromone level to the neighbours according to the group number j differs by one bit.

where *neighbour* is defined as being the group arranged as the ring structure. λ is a pheromone decay parameter and L_{ng} is the length of the shortest route in the neighbour group.

- *Strategy 4*: As shown in Fig. 4, update the pheromone between cities for each group for every R_4 cycles as

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

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

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

- *Strategy 5*: Update the pheromone between cities for each group using both Strategies 1 and 2.
- *Strategy 6*: Update the pheromone between cities for each group using both Strategies 1 and 3.
- *Strategy 7*: Update the pheromone between cities for each group using both Strategies 1 and 4.

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

4. Experimental results and performance study

To evaluate the effectiveness of PACS, we have performed an extensive performance study. In this section, we report our experimental results on comparing PACS with ant system (AS) and ant colony system (ACS). It is shown that PACS and various combinations outperform both ant system (AS) and ant colony system (ACS).

We used three generally available and typical data sets, EIL101, ST70 and TSP225 as the test material [1] to test the performance of the ant system (AS), ant colony system (ACS) and parallel ant colony system (PACS) for the traveling salesman problem.

To ensure a fair comparison among AS, ACS and PACS, 'the number of groups $\times$ the number of ants per group' was kept constant—the number of ants for AS and ACS were set to be 80, one swarm with 80 ants, as reported by 1×80. For PACS, the number of ants was also set to be 80 that was divided into 4 groups with 20 ants in each group (i.e. 4×20) and 8 groups with 10 ants in each group (i.e. 8×10), respectively. The parameters were set to the following values: $\beta = 2$, $q_0 = 0.9$, $\alpha = \rho = \lambda = 0.1$ [3]. The number of iterations for both EIL101 and ST70 were set to be 1000 and TSP225 was set to be 2000 as the cities of TSP225 are more than EIL101 and ST70 data sets. The number of cycles (i.e. R_1, R_2, R_3 and R_4) between updates of the pheromone level from communication for strategies 1–7 in PACS were set to be 30. In order to test the performance of the different approaches to the traveling salesman problem, variously proposed communication strategies for updating the pheromone level between groups in PACS were combined. Where appropriate, these seven communication strategies are applied to the PACS and compared to AS and ACS.

EIL101, ST70 and TSP225 are data sets with 101, 70 and 225 cities, respectively. Experimental results were carried out to the average shortest length for 10 seeds. The performance of PACS (i.e. ACS with communication strategy) is better by in comparison with AS and ACS can be illustrated by Figs. 5–7. As can be seen from Tables 1–3, PACS outperforms both AS and ACS on effectiveness.

The EIL101 data set was used for the first experiment. As shown in Table 1, the average improvement on EIL101 for proposed strategy 5 for 4 groups with 20 ants in each group by comparing with AS and ACS were much better up to be 10.57% and 4.70%, respectively. In comparison with AS and ACS, the average improvement on EIL101 for proposed strategy 3 for 8 groups with 10 ants in each group were 10.41% and 4.52%, respectively.

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

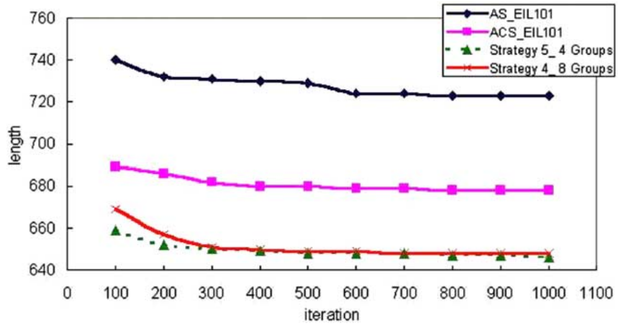

Fig. 5. Performance comparison among AS, ACS and two arbitrarily chosen strategies for EIL101 data set.

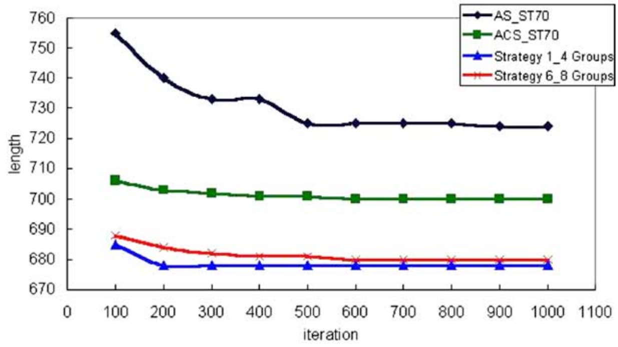

Fig. 6. Performance comparison among AS, ACS and two arbitrarily chosen strategies for ST70 data set.

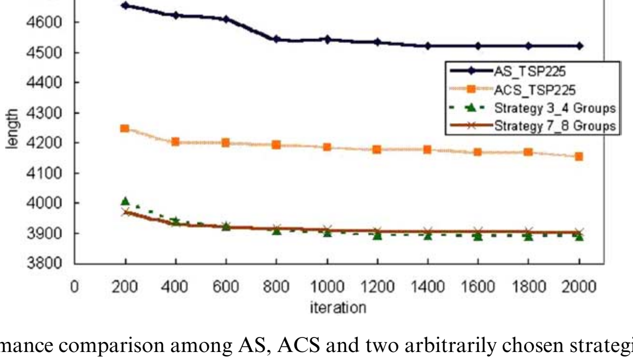

Fig. 7. Performance comparison among AS, ACS and two arbitrarily chosen strategies for TSP225 data set.

Table 1
The performance of ACS with communication strategies (Strategies 1–7) obtained in comparison with AS and ACS for EIL101 data set on TSP problem

Seed	AS	ACS	Strategy 1		Strategy 2		Strategy 3		Strategy 4		Strategy 5		Strategy 6		Strategy 7	
	1, 80	1, 80	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10
1	730	683	657	655	648	653	649	645	654	644	646	651	646	653	647	646
2	730	680	657	655	650	647	643	648	647	649	641	650	660	643	648	643
3	731	681	644	646	655	655	641	646	653	646	641	648	646	645	648	642
4	720	678	645	648	651	654	651	647	647	647	643	646	642	647	650	652
5	727	676	641	643	648	663	648	656	651	651	644	650	647	651	647	647
6	727	673	656	655	648	644	645	655	648	649	647	653	651	653	644	655
7	698	675	642	644	649	658	646	648	651	650	650	646	647	648	645	645
8	726	679	646	651	653	662	658	645	647	645	653	650	647	653	652	655
9	721	672	645	646	651	656	652	642	649	650	652	647	649	651	646	650
10	718	685	643	651	654	647	645	645	652	651	646	647	646	652	650	648
Average	723	678	648	649	651	654	648	648	650	648	646	649	648	650	648	648

Table 2
The performance of ACS with communication strategies (Strategies 1–7) obtained in comparison with AS and ACS for ST70 data set on TSP problem

Seed	AS	ACS	Strategy 1		Strategy 2		Strategy 3		Strategy 4		Strategy 5		Strategy 6		Strategy 7	
	1, 80	1, 80	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10
1	734	701	679	680	683	684	678	683	681	683	684	679	680	677	682	681
2	721	700	681	681	677	686	688	681	677	681	679	681	677	681	677	681
3	722	700	681	682	678	683	678	681	678	681	677	681	681	682	678	677
4	717	701	677	688	682	687	678	682	679	694	685	683	686	681	681	689
5	721	703	678	684	678	686	678	681	678	678	678	679	682	683	677	682
6	713	702	691	690	694	683	678	682	678	678	692	681	689	684	689	686
7	714	700	682	683	678	683	677	683	681	677	681	682	678	677	677	677
8	730	701	677	682	677	679	677	682	681	683	681	677	678	678	678	683
9	730	696	677	681	679	685	678	678	677	682	683	679	683	677	678	681
10	736	699	678	693	682	688	678	681	681	682	680	691	682	678	677	690
Average	724	700	680	684	681	684	679	681	679	682	682	681	682	680	680	683

Table 3
The performance of ACS with communication strategies (Strategies 1–7) obtained in comparison with AS and ACS for TSP225 data set on TSP problem

Seed	AS	ACS	Strategy 1		Strategy 2		Strategy 3		Strategy 4		Strategy 5		Strategy 6		Strategy 7	
	1, 80	1, 80	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10	4, 20	8, 10
1	4587	4145	3907	3933	3913	3914	3879	3885	3903	3949	3882	3905	3866	3905	3884	3943
2	4492	4215	3903	3883	3879	3879	3883	3877	3955	3942	3916	3871	3902	3892	3891	3881
3	4454	4149	3888	3926	3900	3900	3889	3896	3953	3916	3906	3888	3878	3894	3919	3902
4	4609	4160	3892	3886	3908	3952	3889	3885	3895	3890	3871	3885	3866	3879	3919	3899
5	4538	4163	3881	3869	3888	3898	3879	3885	3879	3880	3882	3910	3878	3884	3886	3922
6	4483	4146	3942	3915	3916	3978	3892	3901	3961	3895	3883	3877	3901	3866	3882	3882
7	4555	4149	3904	3911	3876	3939	3881	3891	3881	3887	3881	3876	3892	3885	3882	3912
8	4491	4148	3950	3900	3912	3925	3950	3889	3890	3902	3957	3891	3936	3950	3952	3903
9	4500	4108	3903	3916	3903	3904	3889	3887	3896	3886	3882	3891	3904	3875	3881	3898
10	4521	4161	3877	3915	3875	3911	3877	3896	3876	3873	3884	3876	3919	3917	3895	3909
Average	4523	4154	3905	3905	3897	3920	3891	3889	3909	3902	3894	3887	3894	3895	3899	3905

The ST70 data set was used for the second experiment. As can be seen from Table 2, the average performance of proposed strategy 3 for 4 groups with 20 ants in each group by compared with AS and ACS were 6.20% and 3.06%, respectively and that of proposed strategy 6 for 8 groups with 10 ants in each group were 6.06% and 2.92%, respectively.

The TSP225 data set was also used for the final experiment. Experimental results shown in Table 3, compared with AS and ACS, shows that the average performance of proposed strategy 3 for 4 groups with 20 ants in each group were 13.97% and 6.35%, respectively and that of proposed strategy 5 for 8 groups with 10 ants in each group were 14.06% and 6.44%, respectively.

5. Conclusions

The main contribution of this paper is to propose the parallel formulation for the ant colony system (ACS). Seven communication strategies between groups which can be used to update the pheromone levels are presented. For our preliminary experiments, the proposed parallel ant colony system (PACS) outperforms both ACS and AS based on three available traveling salesman data sets. In general, our presented systems based on data set with large data can get much better performance such that the average improvement of TSP225 is better than that of ST70. The proposed PACS may be applied to solve the quadratic assignment problem [4], data mining [5], space-planning [6], data clustering and the combinatorial optimization problems. We will apply the PACS to data clustering in future.

References

[1] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in: F. Varela, P. Bourgine (Eds.), First Eur. Conference Artificial Life, 1991, pp. 134–142.

[2] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 26 (1) (1996) 29–41.

[3] J.M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.

[4] V. Maniezzo, A. Colorni, The ant system applied to the quadratic assignment problem, IEEE Transactions on Knowledge and Data Engineering 11 (5) (1999) 769–778.

[5] R.S. Parpinelli, H.S. Lopes, A.A. Freitas, Data mining with an ant colony optimization algorithm, IEEE Transactions on Evolutionary Computation 6 (4) (2002) 321–332.

[6] J.A. Bland, Space-planning by ant colony optimization, International Journal of Computer Applications in Technology 12 (6) (1999) 320–328.

[7] M. Dorigo, G.D. Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artificial Life 5 (2) (1999) 137–172.

[8] B. Bullnheimer, G. Kotsis, C. Strauss, Parallelization strategies for the ant system, Technical Report POM 9/97, Institute of Applied Computer Science, University of Vienna, Austria, 1997.
[9] T. Stützle, Parallelization strategies for ant colony optimization, in: Fifth International Conference on Parallel Problem Solving for Nature, Lecture Notes in Computer Science, Vol. 1498, Springer-Verlag, Berlin, Heidelberg, 1998, pp. 722–731.
[10] J.P. Cohoon, S.U. Hegde, W.N. Martine, D. Richards, Punctuated equilibria: a parallel genetic algorithm, in: Second International Conference on Genetic Algorithms, 1987, pp. 148–154.
[11] J.S. Pan, F.R. McInnes, M.A. Jack, Application of parallel genetic algorithm and property of multiple global optima to vq codevector index assignment for noisy channels, Electronics Letters 32 (4) (1996) 296–297.