J Jpn Ind Manage Assoc 59, 449-456, 2009

Original

Ant-TSL System Algorithm using New Ant Agents with Intensification and Diversification Strategies

Taichi Kaji†
Department of Information and Management Science,
Otaru University of Commerce,
3-5-21, Midori, Otaru, 047-8501, Japan

ABSTRACT
Ant system algorithm (AS) proposed by Dorigo and others is a new approach for stochastic combinatorial optimization. They applied the proposed methodology to the classical Traveling Salesman Problem (TSP), and reported simulation results. The results show that the AS for TSP was as effective as tabu search and better than simulated annealing. However, when applying this AS to randomly generated graphs, there is a tendency for the solutions obtained using the AS to be trapped in bad solution. Therefore, we attempt to escape bad solution by improving the original AS, using the following strategies. First, we designed a new agent by using intensification and diversification strategies, such as the tabu search applies, in order to obtain better solutions. And, we tried to solve the problem by using new agents with the ability of local search. Furthermore, the parallel ant system algorithm by the above-mentioned new agents was implemented to reduce computational time. Finally we discuss the characteristics of proposed AS.
Key words: Meta-Heuristics, Ant System Algorithm, Multi Agents, Tabu Search, Parallel Algorithm, Traveling Salesman Problem

1. INTRODUCTION

The ant system algorithm (AS) presented by Dorigo and others [4] is a new computational paradigm, which is a stochastic combinatorial algorithm. Over the past few years, several studies have been made on the AS paradigm [5] [6]. This research is inspired by the problems studied by the ethologists [2][3][12]. For example, it was to understand how ants could manage to establish shortest route paths from their colony to feeding sources and back. It is considered that real ants find the shortest path by a communication process through the pheromone, which is very simple, but exhibit highly structured behaviors and perform complex tasks [10].

The AS inspired by this phenomenon can solve the problem by "ants", that is, agents with very simple, basic capability, which mimic the behavior of real ants. The ants manage to establish shortest paths through the medium that is called "pheromone" which is used to communicate information regarding paths among individuals. The main characteristics of this paradigm are controlled by the combination of positive feedback through pheromone, and greedy heuristic. The greedy heuristic help find a good local solution in the early stages of the search process. The pheromone promotes a global search to escape local solutions by the communication among individual information regarding path.

Dorigo and others applied their methodology to the traveling salesman problem (TSP) and reported the simulation results. It is a well-known fact that several effective algorithms have been made on TSP [9][11]. To compare the AS with some of the effective algorithms in meta-heuristics, they implemented simulated annealing (SA) and tabu search (TS), which are known as meta-heuristics and assessed using Oliver30 for the comparisons. The results of the comparisons on Oliver30 showed that the AS was as effective as tabu search and better than simulated annealing [4].

However, when applying this AS to randomly generated graphs, there is a tendency for the solutions obtained using the AS to be trapped in bad solution. Therefore, we designed a new agent by intensification and diversification strategies, such as the tabu search applies, in order to obtain better solutions. Intensification and diversification are important components of tabu search [7]. We would like to describe how a combination of ant agent and strategies of intensification and diversification could be applied. Furthermore, because the AS does not depend on neighborhood, we apply an approach based on neighborhood to the above-mentioned tabu type of agents to search strongly in local regions of neighborhood. In this proposed algorithm, a unique colony is constructed by the mix of several types of agents from the view of the ability of strong intensification, diversification and local search. This colony from several types of agents leads to improvement in AS. And, we also attempted to reduce the computational time using parallel ant system algorithm by dividing colonies. This type can

† Tel, Fax: +81-134-27-5390
E-mail: tkaji51@res.otaru-uc.ac.jp
Received : August 24, 2007
Accepted : September 4, 2008

be done by partitioning the problem itself into several independent colonies. These algorithms are very well suited for implementation on a MIMD parallel mechanism. Finally, we analyzed the performance of the proposed approaches and compared the standard AS with the proposed ant tabu approach. Herewith, we show that this proposed ant tabu approach gives a better quality of solutions than standard methodology. And, this parallel implementation based on dividing the colony naturally leads us to the conclusion that saves computational time.

The approach by new agents discussed in this paper, designed to "solve" (in the sense of approximating the optimum) the traveling salesman problem, is mathematically unexciting, but has worked remarkably well, both from the point of view of the computational effort involved, and from that of the quality of the solutions obtained on a variety of test problems. And, the agent heuristics approach will become an interesting vehicle in the solving of combinatorial optimization problems.

2. ANT SYSTEM ALGORITHM

The paradigm of ant system algorithm (AS) proposed by Dorigo [4] is inspired by the problem of understanding how almost blind animals like ants could manage to establish shortest route paths from their colony to feeding sources and back [2][3][12]. A moving ant lays some pheromone on the ground. The pheromone is used to communicate information among individuals regarding paths, and used to decide where to go. While an isolated ant moves essentially at random, an ant encountering a previously laid pheromone can detect it and decide with high probability to follow it, thus reinforcing the pheromone on the path with the ant's one. That is, the more ants follow the pheromone, the more attractive to be followed, since the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. And, the quantity of pheromone on the shorter path grows faster than on the longer one. The final result is that very quickly, all ants will choose the shorter path. From this phenomenon, the idea of the AS is induced.

The AS has artificial ants on the computer for the TSP which has n towns and the length d_{ij} of path between towns i and j. Let m be the total number of ants over all towns at time t. Each ant is a simple agent with the following characteristics:

- It chooses the town to go with a probability that is a function of the distance and of the amount of the pheromone information (The ant agent is not blind).
- It can not choose already visited towns until a tour is completed (The ant agent has some memory)
- It attaches the pheromone to the selected path.

Let $\tau_{ij}(t)$ be the intensity of the pheromone information on edge (i, j) at time t. Each ant at time t chooses the next town, where it will be at time t+1. It needs n times to construct a completed tour, and an n time is one cycle (step) of the algorithm. The pheromone information is updated according to the following formula when a tour is completed (i.e., at one cycle).

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

where ρ is a parameter such that (1- ρ) represents the evaporation of pheromone between t and $t+n$. Furthermore $\Delta\tau_{ij}$ is defined as follows;

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \textit{if } k-th \textit{ ant uses edge } (i,j) \textit{ in its tour} \\ 0 & \textit{otherwize} \end{cases} \tag{3}$$

where Q is a given constant value and L_k is the tour length obtained by the k-th ant, and $\Delta\tau_{ij}^{k}$ expresses the pheromone information laid on edge (i,j) by the k-th ant between time t and $t+n$. In other words, the ant lays higher level pheromone on used edges if the tour length of the ant is shorter. Each k-th ant has the tabu list $tabu^k$ that saves the towns visited by k-th ant, in order to forbid choosing the towns already visited between time t and $t+n$. When a tour is completed, the $tabu^k$ is then emptied and the ant is free again to choose. The transition probability from town i and town j for the k-th ant is defined as follows;

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{k\in\{V-tabu^k\}} [\tau_{ik}(t)]^{\alpha}\cdot[\eta_{ik}]^{\beta}} & \textit{if } j \in \{V-tabu^k\} \\ 0 & \textit{otherwise} \end{cases} \tag{4}$$

where τ_{ij} is the pheromone information that guides into a global search, η_{ij} is defined as $1/d_{ij}$ that guides into a greedy heuristic search, and α and β are parameters that control the relative importance of τ_{ij} and η_{ij}. Given the above-mentioned definitions, the Ant System Algorithm is shown as Fig. 1. We call this standard Ant System Algorithm Ant-Standard, proposed by Dorigo and others.

3. THE PERFORMANCE OF THE STANDARD ANTSYSTEM ALGORITHM

Dorigo and others have compared this standard ant system algorithm (Ant-Standard) with other meta-heuristic algorithms. They implemented a simulated

Ant System Algorithm:
t=0 ; $\tau_{ij}(t)$=small positive value c ;
m ants are randomly positioned on the towns;
while (Stop Criterion ≠ True) **do begin**
  while (The tours are not completed) **do begin**
    for k=1 **to** m **do**
      The k-th ant choose the town j with probability
      $p_{ij}^{k}(t)$, and move to the town j form town i ;
  end ;
  for k=1 **to** m **do**
    Compute L_k and $\Delta\tau_{ij}^{k}$;
  Update Pheromone information for every edge(i,j), using equation (1) ;
  t=t+n ;
end.

Fig. 1 Ant System Algorithm.

annealing (SA) and a tabu search (TS), and assessed their performances using the Oliver30 data. The results showed that the Ant-Standard for this problem was as effective as tabu search and better than simulated annealing [4].

However, when applying this Ant-Standard algorithm to randomly generated graphs, where the nodes are randomly laid in the unit square, there is a tendency for the solutions obtained using the Ant-Standard algorithm to be trapped in bad solution. Fig. 2 shows the result for the randomly generated graphs in the range from 100 to 400 nodes, in comparison with simulated annealing. The AS algorithm is analyzed using a parameter of Q=100 and the number of ant agents which is equal to the number of towns, as discussed by Dorigo and others [4]. And, the simulated annealing used a 2-opt neighborhood structure. We have to consider the parameter's values of the simulated annealing in order to obtain good solutions in a reasonable amount of computational time [1]. The initial *tmp* (T) is 500 and the stop *tmp* (T) is 0.01, since, in spite of extending the above range of *tmp*, we did not obtain a significant improvement in the quality of the solutions. We found that *phi*=0.92, which is the cooling schedule parameter, gives an acceptable balance between the solution quality and computational speed. It appears that initial length of the Markov chains r=100 and *tau*=1.1, which control the length of Markov chains, yield good compromises between the quality of the solutions obtained and the time required. The data are average cost, best cost and worst cost obtained after 10 repetitions of both algorithms. The algorithms used here were coded in C++ and implemented on the Intel Celeron 400MHz. We see that a standard ant system algorithm do not produce good results.

Next, Fig. 3 presents traces of run of the Ant-Standard algorithm, which is the length of the best-found tour at each cycle. The result show that the tour cost suddenly drops to low values in early steps, however, after that, the behavior is like the random walk and never converges to optimal solutions.

4. Ant-TSL ALGORITHM

The idea of the ant system algorithm (AS) is very interesting. However, the standard type of the AS cannot obtain better solutions for random graphs. So, we designed a new agent by using intensification and diversification strategies, such as tabu search applies [7], in order to obtain better solutions. Intensifications and diversifications are important components of tabu search. Intensification strategies are based on modifying choice rules to encourage move combinations and solution features historically found good. On the other hand, the diversification stage encourages the search process to examine unvisited regions and to generate solutions that differ in various significant ways from those seen before. We would like to describe how a combination of ant agent and a strategies of intensification and diversification could

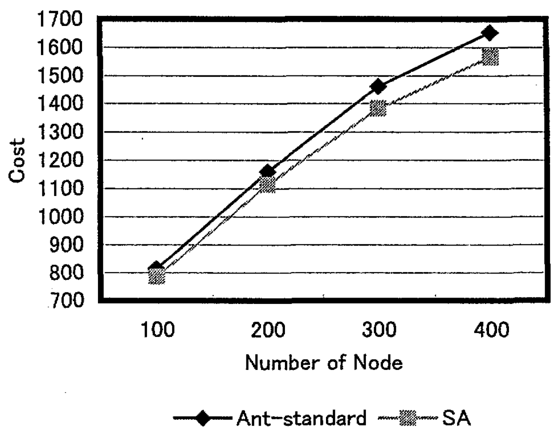

Fig. 2 Comparison of Ant-Standard and SA.

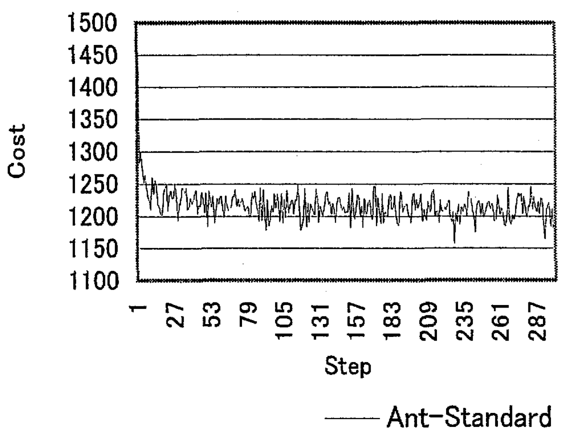

Fig. 3 Trace of Ant-Standard.

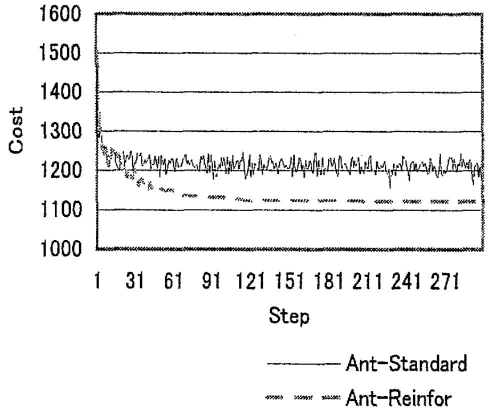

Fig. 4 Trace of Ant-Reinfor (AS with only intensification strategy) compared to Ant-Standard (normal AS).

apply.

First, we have to consider the new agent with the intensification strategy. Generally, the ants in this AS mutually have the communication using the pheromone to find a better tour. The information of the tour obtained by each agent is communicated to the next population of agents through the medium of the pheromone and the next agents can search for a better tour according to this medium. In the Ant-Standard algorithm, the pheromone information is defined as Q/L_k, which gives a higher value if the tour length is shorter.

To analyze the characteristic features of convergence to near optimum for proposed algorithms, we will consider behavior of the obtained solution in response to updating of pheromone in Fig. 3, Fig. 4 and Fig. 5 (Fig. 4 and Fig. 5 are discussed later). Fig. 3 shows that the behavior of tour length cost based on the communication by this way cannot converge, since the information of the best tour in each step is not strongly communicated through the pheromone.

However, there are ideas which effectively use information of pheromone [5][6]. So, we tried to reinforce the pheromone information of the best tour in each step to find a better tour. This also becomes strength of the intensification strategy. That is, the pheromone information laid by an elite agent that obtains the best tour in each step is reinforced as $\sigma \cdot Q/L_k$ to strongly communicate the information of a better tour obtained by an elite agent into the next generation. This intensification strategy is based on modifying the pheromone information of an elite agent to encourage attractive regions to search for better tours. Fig. 4 shows traces of run of best-found solution in each step by the Ant-Standard and the AS with an intensification strategy based on reinforcement of pheromone information, called Ant-Reinfor. The performance of Ant-Standard fluctuates, while the Ant-Reinfor using the reinforcement of pheromone information strongly converges to near optimal solution in the experiment.

The intensification strategy in the AS shows remarkable improvement in contrast with the Ant-Standard. It seems that the reinforcement of the pheromone is an important medium operation of communication. However, using only an intensification strategy, the behavior fell to the situation in which the best found solution in the later steps is similar to the best tour obtained before and it cannot improve in terms of quality of solution, because of the result of convergence strategy using the reinforcement of the pheromone information. This indicates that the system has ceased exploring new possibilities and no better tour will arise. The behavior of the restricted convergence for an intensification strategy is induced by the fact that all agents have only identical ability, which is expressed by equation (4). That is, the behavior of the AS with reinforcement of the pheromone information shows strong convergence in the initial stages by means of the construction of the similar agents; however, the system loses the ability of improvement later on and falls to a stationary situation.

Therefore, we shall attempt to avoid a stationary situation by merging agents with different abilities to the original population to give the system diversification. The new agent for this diversification strategy has a longer-term memory, which is the strategy used in the tabu search approach. The longer-term memory [7] in tabu search is used to diversify the search compelling regions that have not been visited before. Hence, we apply this idea to the ant agents. When either edge (i, j) or (j, i) is used by an ant agent, we increase the longer-term memory element $LMEM[i][j]$; $j>i$ by 1. Let $L_{ij} = C/LMEM[i][j]$, where C is a parameter. We call L_{ij} negative pheromone, which is a signal not to attract agents. The ability of the new agent is formulated as:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta \cdot [L_{ij}]^\gamma}{\sum_{k \in \{V - tabu^k\}} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta \cdot [L_{ik}]^\gamma} & if\ j \in \{V - tabu^k\} \\ 0 & othersize \end{cases} \tag{5}$$

In Fig. 5, we show how the best-found solution in each step decreases for the AS with only an intensification strategy and the AS with both an intensification strategy and a diversification strategy. The results show that the AS with both an intensification strategy and a diversification strategy

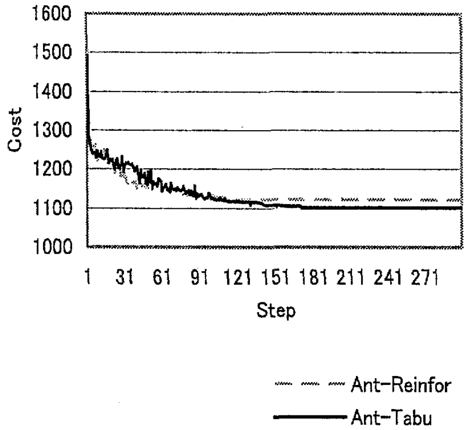

Fig. 5 Trace of Ant-Tabu (AS with both an intensification strategy and a diversification strategy) compared to Ant-Reinfor (AS with only an intensification strategy).

gives a better solution than the AS with only an intensification strategy, by diversification for merging agents with different ability of longer-term memory. We see that a diversification strategy is helpful to obtain better solutions. Thus, the AS algorithm by these agents with the abovementioned intensification strategy and diversification strategy is called the Ant-Tabu algorithm, and this new agent is called the ant-tabu agent.

And, it has been recognized that it is effective to add local search to AS [5][6]. That is, there is a possibility of improvement in terms of quality of solution by giving the ability of local search to the ant-tabu agents, because any solutions obtained by the Ant-Tabu do not generally satisfy the condition for local solution obtained by local search. On the other hand, SA becomes identical to the form of local search in the case where the value of the control parameter is taken near equal to zero. The stop *tmp* for test instances used here is 0.01, and behavior of SA can be viewed as a process of local search as the value of *tmp* approaches 0.01. Therefore, the combination of local search and SA is not effective [1]. So, we attempted to solve the problem using ant-tabu agents with the ability of local search. That is, this ant is a new ant-tabu agent with new characteristics which can search for the best solution in some neighborhood of the tour founded in the colony. However, only agents that find better tours in the colony carry out the local search characteristics. This revised Ant-Tabu system algorithm is called the Ant-TSL system algorithm, and is illustrated in Fig. 6. Thus, the new ant tabu system algorithm has the ability to powerfully search in both global and local regions. And, this Ant-TSL has unique colonies from a mix of several types of agents from the view of the ability of strong intensification, diversification and local search. This colony of several types of agents leads to improvement in AS.

5. COMPUTATIONAL RESULTS AND PARALLEL ANT SYSTEM ALGORITHM

To assess the effectiveness of the proposed Ant-Reinfor, Ant-Tabu and Ant-TSL, we report on some of the experiments. The programs are written in C++ and implemented on the Intel Celeron 400MHz. Table 1 shows the average cost, best cost, worst cost

Ant-TSL Algorithm:
t=0 ; $\tau_{ij}(t)$=small positive value c ;
m ants are randomly positioned in the towns;
while (Stop Criterion ≠ True) **do begin**
  while (The tours are not completed) **do begin**
    for k=1 **to** (int)(m*0.8) **do** // for Ant-Reinfor agents
      The k-th ant choose the town j with probability $p_{ij}^k(t)$ using equation (4),
        and move to the town j form town i ;
    for k=(int)(m*0.8)+1 **to** m **do** // for Ant-Tabu agents
      The k-th ant choose the town j with probability $p_{ij}^k(t)$ using equation (5),
        and move to the town j from town i ;
    Ant agents that find better solutions attempt to solve the problem using local search ability.
      // for Ant-Tabu agents with local search
  end ;
  Reinforce pheromone for elite agents;
  Update longer-term memory and pheromone information using equation (1);
  t=t+n ;
end.

Fig. 6 Ant-TSL Algorithm.

Table 1 Comparison of Ant-TSL with other heuristics on random instance of TSP.

Problem Size	100	200	300	400
Ant-Standard				
Average	815.2	1158.8	1461.2	1652.6
Best	800	1149	1456	1643
Worst	829	1165	1478	1663
Time(s)	109	576	1361	2504
Ant-Reinfor				
Average	796.8	1116.0	1377.4	1573.0
Best	783	1107	1366	1554
Worst	814	1125	1394	1594
Time(s)	108	570	1353	2457
Ant-Tabu				
Average	781	1099.8	1369.4	1560.4
Best	778	1094	1356	1533
Worst	786	1105	1380	1577
Time(s)	114	625	1701	9192
Ant-TSL				
Average	774	1091.5	1338.4	1515.2
Best	774	1088	1336	1512
Worst	774	1095	1342	1517
Time(s)	128	594	1406	2534
SA				
Average	786	1113	1381.2	1565.2
Best	774	1099	1376	1546
Worst	798	1127	1389	1577
Time(s)	593	598	637	677

and CPU time obtained after 10 repetitions of Ant-Standard, Ant-Reinfor, Ant-Tabu, Ant-TSL and simulated annealing (SA) on the generated random graphs. The nodes of generated random graphs are points whose two coordinates are each real numbers chosen randomly from interval [0, 100], with instance sizes from 100 to 400. In the experiments in this paper, numerical parameters are set to the following values: α=1.0, β=5.0, γ=2.0, C=10, Q=100. The number of iteration until stop criterion is satisfied is 300. These values were obtained by a preliminary optimization phase, which found that the experimental optimal values of the parameters were largely independent of the problems in the range of size for this test instance used here. And the number of ant agents is equal to the number of cities, although Ant-TSL uses only 100 agents. In Ant-Tabu and Ant-TSL, the ant-tabu agents with longer-term occupy 20% of colony. On the other hand, the parameters of SA are discussed in detail in Section 3, and similarly they are set to the following values; the initial *tmp* is 500, the stop *tmp* is 0.01, *phi*=0.92, *r*=100, *tau*=1.1.

First, we see that Ant-Reinfor and Ant-Tabu outperforms the Ant-Standard in terms of solution quality, and Ant-Tabu is better than Ant-Reinfor. The Ant-Tabu with intensification and diversification strategies offers a slightly better performance than SA. From this point, we can see that the combination of ant agent and the strategies of intensification and diversification used in the tabu search is effective to obtain better solutions. Ant-TSL using ant-tabu agents with the local search system, gives a better solution than the Ant-Tabu and others. And the cost difference between the best cost and the worst cost is close. It is remarkable that Ant-TSL, in spite of a little number of agents, outperforms the standard ant system algorithm, Ant-Tabu and SA in terms of quality of solution. Our results indicate that the proposed colony from a mix of several types of agents from the view of the ability of intensification, diversification and local search leads to improvement in AS.

Moreover, we consider the relative error for optimum to measure the relative phase of the quality solution obtained by Ant-TSL. For this purpose, we report test problems in TSPLIB with their optimal solution values in Table 2. Table 2 shows the average cost over 10 trials, the optimum and the relative error. Ant-TSL offers the better performance with respect to relative error, and only for the kroA150 problem it finds a slightly worse solution. Of course, it is a well-known fact that the Lin-Kernighan variant methods obtain the best quality of solution in TSP. Ant-TSL is just as good as the Lin-Kernighan variant in this case, and it is comparable to the other outstanding heuristics. From these results, we may say that Multi-

Table 2 Results obtained by Ant-TSL on TSP problem in TSPLIB.

	average	optimum	relative error(%)
eli51	426.2	426	0.05
st70	675.0	675	0.0
eli76	538.0	538	0.0
KroA100	21282.0	21282	0.0
KroA150	26619.2	26524	0.3
KroA200	29383.0	29368	0.05

Agent techniques like AS are the great potential areas of development in the future.

Next, we also must evaluate the performance for CPU time of the proposed algorithm. The ant system algorithm (AS) is based on the mass of agents in a colony. Accordingly, the AS algorithm is time-consuming as shown in Table 1 because of processes of a mass of agents. It is like the genetic algorithm, which is also time-consuming because of computing the numbers of genes. This is serious weak point in the AS algorithm. Therefore, we attempted to reduce the computational time using parallel ant system algorithm by dividing colonies. This type can be done by partitioning the problem itself into several independent colonies. The algorithm is very well suited for implementation on a MIMD parallel mechanism. The parallel algorithm has been implemented on a network by MPI library [8]. And, the computation by agent can be parallelized. This will generally require the partition of a set of the colonies into P processors. Each of these divided colonies is assigned to one process, which computes m/P tours. Next, the pheromone information of sub colonies is updated. That is, we divide the set of ant agents C into $C_1, C_2, \ldots, C_p$ such that $|C_i| = m/P$ for all i, $C_1 \cup C_2 \cup \ldots \cup C_p = C$, and $C_i \cap C_j = \emptyset$, for $i \neq j$, where P is the number of processor. Each partitioned colony is assigned to each processor, which computes tours by ant agents in the C_i. After computing m/P tours, the pheromone information of each colony is determined and mutually communicated among all processors together. The synchronization is required at each iteration step. Normally, this technique requires extensive communication since the synchronization is required at each cycle. However, the communication of parallel AS algorithm can be constructed using only the pheromone information, without other information (i.e., a number of tours of agents in a sub colony). It is worth applying to only the problem in which the computation in one step is complex and time-consuming. So, we estimate that this proposed parallel algorithm allows us to considerably reduce the computational time.

We evaluate the performance of the parallel ant system algorithm using 100 agents used in the Ant-TSL, with the number of nodes ranging from 100 to 400. Fig. 7 illustrates the trace of CPU time required for sequential AS (cpu 1 line), two processors (cpu 2 line) and four processors (cpu 4 line) for the parallel AS. The computational time required for the SA is about 600sec and does not remarkably increase. Although, the time required for the AS algorithm is about 110sec in the case of 100 nodes, the rate of an increase for computational time is high (see cpu1 line in Fig. 7) because of the computation of a mass of agents in a colony. However, in Fig. 7 we see that the parallel AS can reduce computational time lineally as the number of processors increase. Running time is nearly 1/2 and 1/4 of CPU time of sequential AS for the number of processor. There are various types of AS by several kinds of strategy and parameters, and the running time is distinct. However, other AS can also reduce computational time lineally as the number of processors. It should be noted that this parallelism for the AS allows us to reduce considerably the CPU time. It is possible that the AS obtain better solutions in reasonable computational time by parallel colonies.

6. CONCLUSIONS

The ant system algorithm (AS) proposed by Dorigo is a unique heuristics approach, which is the search activities over so-called ants, that is, agents with very simple basic capabilities which, to some extent, mimic the behavior of real ants. However, this standard type of the AS cannot obtain better solutions for TSP on random graphs. So, we proposed new agents with the ability of intensification and diversification strategies, which are highly important components of the tabu search. We illustrated the combination of ant agents and the strategies of intensification and diversification. We showed revised AS by combining these ant agents of tabu search type and local search to search strongly in local regions. It is helpful to apply the idea based on neighborhood to ant agents in terms of improving the

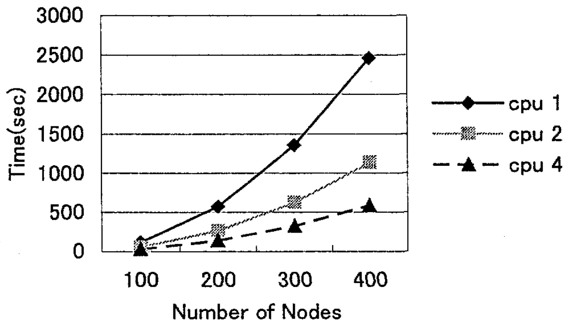

Fig. 7 CPU Time for Parallel Ant System Algorithm.

quality of solutions because the AS does not depend on neighborhood such as 2-opt. The computational results show that proposed Ant-TSL algorithms outperform the standard ant system algorithm, the Ant-Tabu and the SA in terms of solution quality. We show that intensification and diversification strategies, and the ability of local search in ant agents are helpful to obtain better solutions.

Next, we also considered CPU time for the AS which requires much more computational time. Therefore, we examined the parallel ant system algorithm by partitioning the colony itself into several independent colonies. This algorithm is well suited for parallel computation. It can communicate among each colony using only signals of pheromone information. Numerical results show that parallel ant system algorithm can reduce computational time lineally as the number of processors increase. It is possible that the AS obtain better solutions in reasonable computational time by parallel colonies. In summary, our results indicate that the proposed Ant-TSL system algorithm that works together with several types of agents can be used to obtain good-quality approximate solutions for a wide variety of TSP.

REFERENCES

[1] Ararts, E. and Korst, J. : Simulated Annealing and Boltzmann Machines, John Wiley & Sons (1989)

[2] Deneubourg, J.L., Pasteels, J.M., and Verhaeghe, J.C. : "Probabilistic Behavior in Ants: A Strategy of Errors?," *J. Theor. Biol.*, Vol. 105, pp. 259-271 (1983)

[3] Deneubourg, J.L. and Goss, S. : "Collective Patterns and Decision-making," *Ethol., Ecol. Evol.*, Vol. 1, pp. 295-311 (1989)

[4] Dorigo, M., Maniezzo, V., and Colorni, A. : "The Ant System : Optimization by a Colony of Cooperating Agents," *IEEE Trans. Syst., Man Cybern. Part B*, Vol. 26, No. 1, pp. 1-26 (1996)

[5] Dorigo, M. and Gambardella, L.M. : "Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Trans, Evol. Comput.*, Vol. 1, pp. 53-66 (1997)

[6] Dorigo, M. and Stutzle T. : Ant Colony Optimization, The MIT Press (2004)

[7] Glover, F. and Laguna, M. : Tabu Search, Kluwer Academic Publishers (1997)

[8] Gropp, W., Lusk, E., and Skjellum, A. : Using MPI, The MIT Press (1999)

[9] Gutin, G. and Punnen, A.P. : The Traveling Salesman Problem and Its Variations, Kluwer (2002)

[10] Kawamura, H., Yamamoto, M., Mitamura, T., Suzuki, K., and Ohuchi, A. : "Cooperative Search Based on Pheromone Communication for Vehicle Routing Problems," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, Vol. E81-A, No. 6, pp. 1089-1096 (1998)

[11] Lawler, E.L., Lenstra, J.K., Rinnooy Kan A.H.G., and Shmoys, D.B. : The Traveling Salesman Problem, John Wiley & Sons (1985)

[12] Nakamura, M. and Kurumatani, K. : "A Mathematical Model for the Foraging of an Ant Colony and Pattern Formation of the Pheromone Trial, Fundamental Theories of Deterministic and Stochastic Models in Mathematical Biology," *Inst. Stat. Math. Coop. Res. Rep.*, vol. 76, pp. 120-131 (1995)