Comparison of Some Migration Policies in Sub-Ant-Colony Scheme

Tzung-Pei Hong
Dept. of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan
tphong@nuk.edu.tw

Ling-I Huang
Dept. of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan
m1015511@mail.nuk.edu.tw

Wen-Yang Lin
Dept. of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan
wylin@nuk.edu.tw

Abstract
In the past, many approaches were proposed to improve the diversity and thus the quality of the solution in bio-inspired computation. In this paper, we discuss the issue in the ant-colony-system. We consider partitioning an ant colony into several sub-ant-colonies with the same total number of ants and the migration effects among them. Several migration strategies are adopted and compared. Experiments conducted on a TSP data show the results with migration is in general better than those without migration.

Keywords—Bio-inspired Computation; Ant Colony System; Sub-ant-colony; Migration

I. INTRODUCTION
Bio-inspired computation is a new field derived from the behavior of living things and has been widely applied to speed up the execution of a variety of applications. There have been many approaches proposed in the field, and among them the ant colony system [2] gets good solutions for optimization problems.

In the past, many approaches were proposed to improve the diversity and thus the quality of the solution in bio-inspired computation. For example, multi-population genetic algorithms (MGAs) [12] have been recognized as being efficient and effective in regard to finding nearly optimal solutions. Multi-colony ant systems are also beneficial to optimization problems [14][16]. An advantage of these approaches is to shorten the number of generations needed to find optimal solutions. They can find more than one local optimal solutions and thus has a higher probability to resist premature convergence.

In the paper, we discuss several migration policies in the sub-ant-colony scheme which partitions an ant colony into several sub-ant-colonies with the same number of ants. The sum of the numbers of the ants in all the sub-ant-colonies is equal to the original ant number of the whole colony. The performance of the migration policies is also compared from experiments with different settings. The policies include the best-tour migration policy, the weighted best-tour migration policy, the conditional weighted best-tour migration policy by edge, and the conditional weighted best-tour migration policy by tour. The results show that the migration can improve the solution quality.

II. REVIEW OF RELATED WORKS
Ant Colony Optimization (ACO) is a meta-heuristic approach proposed by Dorigo et al. [2][4][7][8]. In the traditional Ant Colony Optimization Algorithm, a transferred graph in which the edges are all endowed with a weighted value is set according to the given optimization problem. Artificial ants aim to work on the weighted graph to find the optimal solution. The weighted graph contains flexible information called pheromone laid by artificial ants. The value of the pheromone can be altered by a pheromone updating mechanism. Dorigo and Gambardella [6] then proposed Ant Colony Systems (ACS) as a derivative method of Ant System (AS) [8]. There were also several variants of ACS proposed for improving performance and for different applications [10][11][13].

The ACS designed a specific transition probability to randomly decide the next node to visit from the current node. Besides, two kinds of pheromone update processes are performed, local update and global update. The local pheromone update is performed whenever each ant makes a move until it forms a path. It will avoid the solution being locally blocked in a small range or converging to some certain paths. The other update is the global pheromone update, which is performed after a complete path is finished. The pheromone amounts of only the edges of the best solution are updated.

Manfrin et al. considered the migration effect in a ring structure using a simple migration policy [14]. In this paper, we discussed and compared the effects of different migration policies.
III. MIGRATION POLICIES

There may be different connection ways for sub-ant-colonies to migrate. In this paper, the connection topology for sub-ant-colonies is the ring structure shown in Figure 1. Each sub-ant-colony has its own identifier number. The ring is directed, meaning it has only one direction. Migration occurs from one sub-ant-colony to its next neighboring one according to the direction. Several migration strategies are adopted to implement the sub-ant-colony scheme. Each sub-ant-colony individually runs the ACS algorithm and when a predefined number of generations are reached, the best solution in each sub-ant-colony is migrated to the other sub-ant-colonies.

![Ring Structure](image)

Fig. 1. The ring structure adopted in the migration scheme.

A flexible weight $w$ distributed over $[0, 1]$ can affect the choice of migration strategy. The migration procedure is performed at the best solution in each sub-colony so far when migration occurs. Each migration strategy is described as follow.

(a) Best-Tour Migration Policy

This policy asks a sub-ant-colony to migrate the pheromone of its best tour so far to its next directed neighbor every fixed number of intervals. The neighboring sub-ant-colony then replaces the amounts of pheromone on the corresponding edges of the best tour with the migrated amounts of pheromone.

(b) Weighted Best-Tour Migration Policy

This policy is similar to the best-tour migration policy. It asks a sub-ant-colony to migrate the pheromone of its best tour so far to its next directed neighbor every fixed number of intervals. The neighboring sub-ant-colony then modifies the amounts of pheromone on the corresponding edges of the best tour in a weighted average way. Formally, assume $w$ denotes the weight, $0 \leq w \leq 1$, and $i$ is an edge on the best tour. The pheromone on edge $i$ is changed according to the following formula:

$$\tau'_{[i+1] \mod c} = w \cdot \tau_{\text{BEST}s} + (1 - w) \cdot \tau_{[i+1] \mod c},$$

where $\tau'_{[i+1] \mod c}$ represents the pheromone amount on edge $i$ in sub-ant-colony $C_{[i+1] \mod c}$, which is the next directed neighbor of the sub-ant-colony $C_s$. $\tau'_{\text{BEST}s}$ is the pheromone amount of edge $i$ of the so-far best solution $\text{BEST}_s$ in $C_s$, and $w$ is the weight mentioned above. Note that the pheromone on an edge not on the best tour will not be changed.

When the weight $w$ is equal to one, the policy will use the amounts of pheromone of the edges on the best solution in a sub-ant-colony to completely replace those in its neighbor; When $w$ is equal to zero, it will degenerate to no migration. The value of $w$ can be set within $0$ to $1$ to represent an appropriate fusion effect between the original pherome and the migrated pheromone.

An example is given below to illustrate the policy. Assume the best solution in sub-ant-colony $C_1$ is 3-2-5-1-4-3, and the value of pheromone of edge (3, 2) is 0.0233. The value of pheromone of edge (3, 2) in the neighboring sub-ant-colony $C_2$ before migration is 0.011. If $w = 0.5$, the new value of pheromone of edge (3, 2) in $C_2$ becomes $0.5 \cdot 0.0233 + 0.5 \cdot 0.011$, which is 0.01715. If $w = 1$, then the new pheromone value in $C_2$ is 0.0233, which is the same as the pheromone value in $C_1$. If $w = 0$, then the new pheromone value in $C_2$ is still 0.011 without change.

(c) Conditional Weighted Best-Tour Migration Policy by Edge

The policy is similar to the above one in addition to one more condition. It will compare the pheromone amount of each edge in the best solution of a sub-ant-colony with that of the corresponding edge in the neighbor. If the former is greater than the latter, then migration occurs and the average pheromone fusion mentioned above is done; otherwise, no migration is done. For the example above, since the pheromone amount on the migrated edge is 0.0233, greater than the original pheromone amount (0.011), the weighted average is then calculated as 0.01715. If $w = 1$, then the policy will degenerate to the conditional best-tour migration policy by edges.

(d) Conditional Weighted Best-Tour Migration Policy by Tour

This policy is similar to the second one except for different migration conditions. In the second policy, the comparison is done on the basis of edges, but here the comparison is based on tours. Restated, if the length (quality) of the best solution in a sub-ant-colony is shorter (better) than that of its neighbor, the above weighted migration occurs; otherwise, no migration is done.

Take the above example to illustrate it. The length of the best solution (which is 3-2-5-1-4-3) in $C_1$ is 15.8 and the length of the best solution in $C_2$ is 16.0. The condition ($15.8 < 16.0$) is satisfied and migration occurs. If $w = 1$, then the policy will degenerate to the conditional best-tour migration policy by tours.

IV. THE ALGORITHM

With the above policies, the proposed algorithm for the ant colony system first partitions the whole
colony into several sub-ant-colonies, and then each sub-ant-colony gets its own best solution. Each sub-ant-colony then migrates the best solution to its neighbor whenever the migration interval is reached. The algorithm is described in details below.

The Sub-Ant-Colony Migration Algorithm for ACS:

**INPUT:** A total size \( x \) of ants, a sub-ant-colony number \( c \), a transferred graph \( G \) with \( n \) nodes for the problem to be solved, an initial pheromone amount \( T_0 \), an iteration number \( t \), and a migration interval \( m \).

**OUTPUT:** A nearly optimal solution for the transferred graph \( G \).

**STEP 1:** Divide the total \( x \) ants into \( c \) equal-sized sub-ant-colonies, each of which thus has \( \frac{x}{c} \) or \( \lceil \frac{x}{c} \rceil + 1 \) ants.

**STEP 2:** Initialize the pheromone amount on all edges of a graph \( G \) as \( T_0 \).

**STEP 3:** Assign each \( s \)-th sub-ant-colony \( C_s \), for \( s = 1 \) to \( c \), to work individually on the transferred graph \( G \) by the following substeps.

**STEP 3.1:** Randomly put each \( r \)-th ant \( a_{sr} \) in the sub-ant-colony \( C_s \), on a node of \( G \) for \( r = 1 \) to \( |C_s| \), where \( |C_s| \) is the number of ants in \( C_s \).

**STEP 3.2:** Set the so-far best solution \( BEST_s \) for \( C_s \) as null with length being \( \infty \).

**STEP 3.3:** Set \( j = 1 \) and \( k = 1 \), where \( k \) is used to represent the current iteration number and \( j \) is used to represent the iteration number after a migration operation is done.

**STEP 3.4:** Move each ant \( a_{sr} \) to the next city by the transition probability rule mentioned before, which decides the edge for \( a_{sr} \) to pass.

**STEP 3.5:** Update the pheromone value on the selected edge by using the local updating process mentioned after an ant \( a_{sr} \) walks through it. If each ant \( a_{sr} \) has not found a feasible solution (completed a tour), go to **STEP 3.4** for continuously selecting the next edge; otherwise, do the next step.

**STEP 3.6:** After all the ants in \( C_s \) find feasible solutions, choose the best solution \( A_s \) from them and execute the global updating process mentioned before.

**STEP 3.7:** Compare \( A_s \) with \( BEST_s \). If \( A_s \) is better than \( BEST_s \) (null value is the worst), set \( BEST_s = A_s \); otherwise, do nothing.

**STEP 3.8:** If \( j = m \), set \( j = 0 \), migrate \( BEST_s \) in each \( C_s \) to its next sub-ant-colony \( C_{[s+1] \mod c} \), and replace the pheromone amount \( t' \) of each edge \( i \) of \( BEST_s \) in \( C_{[s+1] \mod c} \) by the adopted migration policy.

**STEP 3.9:** If the iteration number \( k < t \), set \( k = k + 1 \), \( j = j + 1 \), and repeat **STEPS 3.4 to 3.9**; otherwise, do **STEP 4**.

**STEP 4:** Compare all \( BEST_s \) for \( s = 1 \) to \( m \) to find the best one, called \( ALLBEST \), among them.

**STEP 5:** Output \( ALLBEST \) as the solution of the problem to be solved.

Note that the above strategies and algorithm can also be easily extended to migrate multiple solutions at one time from one sub-ant-colony to another sub-ant-colony.

**V. EXPERIMENTAL RESULTS**

In this section, the experimental results are given to compare the performance of the different migration policies in the sub-ant colony algorithm. The experiments were implemented in the C language on a personal computer with Intel Core i5 CPU 661 with 3.33GHz and 1.8GB RAM. In the experiments, the travelling salesman problem (TSP) was used for implementation. A data called Kroa100.tsp (with 100 cities) from Library of Traveling Salesman Problems [17] was used in the experiments. In the data, the longitude and latitude of the nodes were recorded to form the map. Totally 200 ants were used in the experiments. The parameters \( \alpha \) and \( \beta \) of the transition rule were set at 1 and 2, the probability parameter \( q_0 \) was set as 0.75, and the evaporation rate \( \rho \) was set at 0.8.

Experiments were made to show the efficiency of the adopted migration policies for the migration interval being 25. The results are shown in Figures 2 to 4 for different policies. In these figures, strategy (b) represents the weighted best-tour migration policy, strategy (c) represents the conditional weighted best-tour migration policy by edge, and strategy (d) represents the conditional weighted best-tour migration policy by tour.

![Fig. 2. The average solution quality for migration strategy (b).](image-url)
Besides, there is no significant difference for the final minimum length obtained from the three policies. The policy (d) with $w = 0.5$ is the best among these lines. It is only a little better than the others.

VI. CONCLUSION AND FUTURE WORK

In this study, we have adopted several migration policies in the sub-ant-colony scheme and compared their performance from experiments with different parameter settings. Experimental results on a TSP dataset show that the results with migration are in general better than those without migration. In the future, more experiments will be conducted to further show the characteristics and behavior of migration for different parameters and values.

REFERENCES

The Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence, 2006.

