# Stagnation Control Using Interacted Multiple Ant Colonies

**Alaa Al-Janaby[a], Ku Ruhana Ku Mahamud[b], Norita Norwawi[c]**

*College of Arts and Sciences*
*University Utara Malaysia, 06010 Sintok, Kedah*
*[a]E-mail: alaa.aljanaby@gmail.com [b]E-mail: ruhana@uum.edu.my [c]E-mail: nmn@uum.edu.my*

**ABSTRACT**

*Stagnation is a common problem that all ant algorithms suffer from regardless of their application domain. This paper proposes a new algorithmic approach that can effectively be used to tackle combinatorial optimization problems with a good chance to control the stagnation. The new approach utilizes multiple ant colonies with certain techniques that efficiently organize the work of these colonies to avoid the stagnation situations. Computational tests show that the proposed approach is competitive with other state of art ant algorithms.*

**Keywords**
*Ant Colony Optimization, Combinatorial Optimization Problems, Stagnation.*

## 1.0 INTRODUCTION

Ant Colony Optimization (ACO) is a biological inspiration simulating the ability of the real ant colony of finding the shortest path between the nest and the food source. The main element of ACO success is the use of a combination of priori information (heuristics) about the quality of candidate solutions (also called greedy strategy) and posteriori information (pheromone) about the goodness of the previously obtained solutions.

The problems tackled by ACO are called combinatorial optimization problems. These complex problems arise when the task is to find the best out of many possible solutions to a given problem, provided that a clear notion of solution quality exists. (Dorigo et al., 2006). Traveling salesman problem (TSP), quadratic assignment problem, vehicle routing problem, job secluding problem and network routing problem are some well known examples of these problems.

Number of ACO algorithms is available in the literature. Ant System (Dorigo et al., 1996), Ant Colony System - ACS (Dorigo & Gambardella, 1997), Max-Min Ant System - MMAS (Stützle & Hoos, 2000), Ranked Ant System - RAS (Bullnheimer et al., 1999) and Best Worst Ant System – BWAS (Gordon et al., 2002) are well known ACO algorithms. These algorithms show interesting performance. However, these algorithms are still far from being ideal, these algorithms can get a good solution at the early stages of the algorithm execution but unfortunately all ants speedily converged to a one solution and then the algorithm is unable to improve that solution. This is a common problem that all ACO algorithms suffer from regardless of the application domain; it is called search stagnation problem. The chance of stagnation proportionally increases with the increase of the problem size.

This paper proposes the control of stagnation problem using a new algorithmic approach that utilizes multiple ants' colonies with certain techniques to organize the activities of these colonies. The rest of this paper is organized as follows. Section 2 reviews the related literature. Section 3 proposes the new algorithmic approach. The computational results are presented in section 4. Section 5 concludes the paper and suggests the future work.

## 2.0 LITRURATE REVIEW

There are number of attempts to control or mitigate the search stagnation. The first attempts were in the early wok of Dorigo, Maniezzo, and Colorni (1997). They added a pheromone evaporation mechanism to the ant system the first ACO algorithm. Evaporation is an example of pheromone control which reduces the amount of pheromone on all paths to prevent the high pheromone concentration on the optimal path.

Pheromone limiting (Stützle & Hoos, 2000) is another way of pheromone control which puts upper and lower limits on the amount of pheromone on each path. The upper limits prevents the generating of dominant path and the lower limit prevents the amount of pheromone on each of being zero (never lose the chance of being selected). However, the pheromone limiting will have no effect when the pheromone concentration on a path reaches the max limits. Pheromone limiting must be used in conjunction with evaporation. Evaporation and pheromone limiting are not suitable to be used for static combinatorial problems. The optimal paths are not changed in such problems. Therefore, the application of evaporation may direct the ants to non-optimal paths.

An attempt was done using multiple ant colonies algorithm by Kawamura et al. (2000). The algorithm used a large number of parameters that must be set in advance. These parameters determine the effect of each colony to all other colonies and they organized as an array of size M×M, where M is the number of colonies. No specific way of choosing this large number of parameters was shown. The algorithm tested on some TSP instances and the results were comparable with AS results but can not outperform the results of the best known ant algorithms like ACS and MMAS.

Sim and Sun (2003) propose some conceptual ideas of MACO approach as a new ACO framework for network routing problem. The authors believe that using multiple ant colonies to explore the network offers the opportunity to find new and better paths and reduces the chance of stagnation.

## 3.0 THE PROPOSED APPROACH

Figure 1 shows the basic components of the proposed approach, in this approach there are three mechanisms that are used to organize the work of the individuals in each colony and the work of all colonies. In other words there are two levels of interaction the first one is the colony level and the second one is the population level.

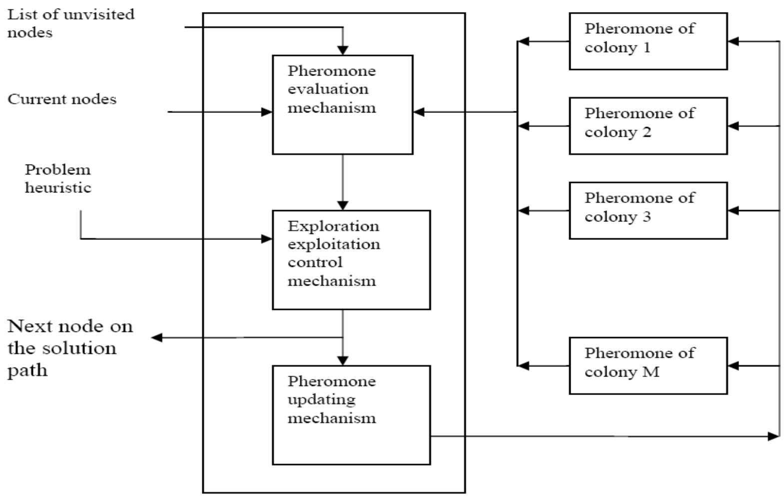


*Figure 1:* The proposed approach

The colony level interaction can be achieved through the pheromone depositing process within the same colony; the pheromone updating mechanism is responsible for the implementation of this kind of interaction. The population level interaction is achieved by evaluating the pheromones of different colonies using some evaluation function; the responsibility here is of the pheromone evaluating mechanism. The degree of interaction of the different colonies is the role of the exploration / exploitation mechanism. This algorithmic approach will be called hereafter Interacted Multiple Ant Colonies Optimization (IMACO).

The work activities of a single colony in the proposed IMACO algorithm are based on ACS. Each colony has its own pheromone that is used as an interaction between the ants of the same colony. The interaction between ant colonies using pheromone can be organized in different terms. The IMACO algorithm is described as follows. *M* colonies of *m* ants each are working together to solve some combinatorial problem. The probabilistic decision of the ant *k* belongs to the colony *v* to move from node *i* to node *j* is defined as:

$$j = \begin{cases} \arg \max_{l \in N_i^{kv}} \{ f(P_{il})\, H_{il}^{\beta} \} & if \;\; q \le q_0 \\ S & \text{otherwise} \end{cases} \qquad (1)$$

$f(P_{ij})$ is the evaluation function of pheromone on the edge ($i$, $j$). and $H_{ij}$ is the problem dependent heuristic. $N_i^{kv}$ is the set of remaining nodes to be visited by the $k^{th}$ ant of colony $v$. $\beta$ is a parameter that determines the relative importance of pheromone versus heuristic, $q$ is a random variable distributed in [0, 1] and $q_0$ is a parameter and $0 \le q_0 \le 1$. $S$ is a random variable selected according to the following probabilistic rule.

$$S = \begin{cases} \dfrac{f(P_{ij})\, {H_{ij}}^{\beta}}{\sum_{l \in N_i^k} f(P_{il})\, {H_{il}}^{\beta}} & \text{if } j \in N_i^{kv} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

### 3.1 Pheromone evaluation mechanism

Two kinds of pheromone evaluation are proposed in this paper. The first one is evaluating the pheromone as an average of the pheromone values of all colonies on some edge. This means that an ant will make its decision to choose some edge based on the average of the available experiences of ants of all colonies that visited this edge in the past. This variant of IMACO is referred hereafter as IMACO-AVG.

Given that for each edge there are *M* pheromone values each belongs to a single colony. Average pheromone evaluation function evaluates the pheromone on any edge as an average of the available *M* values. The pheromone evaluation function for IMACO-AVG will be defined as:

$$f(P_{ij}) = \frac{\sum_{v=1}^{M} P_{ij}^{v}}{M} \quad (3)$$

$P_{ij}^{v}$ is the pheromone of colony *v* on the edge (*i*,*j*). The other mechanism evaluates the pheromone as the maximum value of the pheromone values of all colonies on some edge. In this variant, referred as IMACO-MAX, an ant's decision to choose some edge will be based on the best available experience of ants of all colonies that previously visited this edge. This mechanism chooses the max value among the available *M* values. The pheromone evaluation function for IMACO-MAX is defined as:

$$f(P_{ij}) = \underset{v=1}{\overset{M}{Max}} \; P_{ij}^{v} \quad (4)$$

### 3.2 Exploration / exploitation control mechanism

Each ant makes a probabilistic decision when it needs to move to a new node. The probabilistic decision is based on heuristic information (cost) and pheromone information. Pheromone represents information about previous experiences of the ant's own colony and of the other colonies. While heuristic represent a priori information about the goodness of a solution. The relative importance of heuristic and pheromone information is determined by using the weighting parameter $\beta$.

Another parameter $q_0$ is usually used in ant's probabilistic decision as trade-off between exploitation (choosing the edge with the higher value of the multiplication of pheromone and heuristic values) and exploration (choosing the edge randomly according to some probability distribution).

In this paper we set $\beta=2$ and $q_0=.9$ which are the common values used in ACS single colony algorithm. However, the authors currently conducting a research work that is experimentally testing the cases were different values be used by different ant colonies for these two parameters. The goal of these tests will to reach a balanced form of exploitation / exploration that yields to the best algorithm performance.

### 3.3 Pheromone updating mechanism

The proposed pheromone updating mechanism encourages a balanced form of exploitation of previous experiences and exploration of new or improved paths. Basically the mechanism incorporates global and local pheromone updating. Local pheromone update encourages the exploration of new areas of the search space by reducing the importance of the visited edges. While, global pheromone update encourages the exploitation of previously good solutions by giving extra weight to the edges of global best solutions.

Global pheromone updating includes that best ant of each colony deposits an amount of pheromone on its own path. The best ant refers to the ant that got the so far best solution since the starting of the algorithm execution. After all ants of all colonies complete their tours (i.e., one algorithm iteration), the ant that finds the so far best solution in its colony will be allowed to deposit an amount of the colony's pheromone on the edges of its tour according to the following rule:

$$P_{ij}^{v} = (1-\sigma)P_{ij}^{v} + \sigma \Delta p_{ij}^{v.bs} \quad (5)$$

Where σ is the trail evaporation such that (1- σ) represents the pheromone persistence. This parameter is used to avoid unlimited accumulation of pheromone trails and allows the algorithm to forget previously done bad choices. $\Delta P_{ij}^{v.bs}$ is the pheromone quantity added to the connection (*i*, *j*) belonging to the best solution of $v^{th}$ colony $L^{v.bs}$ and is given by:

$$\Delta p_{ij}^{v.bs} = \begin{cases} 1/L^{v.bs} & if\ (i,j) \text{ belongs to the best tour of colony } v \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Local pheromone update is then applied by each ant on the visited edges. It includes that each ants reduces the amount of pheromone on paths it uses in order to give a more chance to other paths to be chosen by the future generations. It is very important rule as it is performed during the solution construction this helps to yield different pheromone evaluation values for the same edge in the same iteration at different solution construction steps and it is given by:

$$P_{ij}^{v} = (1-\gamma)P_{ij}^{v} + \gamma p_0 \qquad (7)$$

Where $\gamma$ is another pheromone evaporation parameter and $P_0$ is the initial pheromone value.

## 4.0 COMPUTATIONAL RESULT

IMACO-AVG and IMACO-MAX have been tested using two TSP benchmark instances which are kroA100 and lin318 taken from TSP library. The optimal solution for kroA100 is 21282 and that for lin318 is 42029. Number of experiments was run using 1, 2, 3, 4, 5, 7, 9, 12, 15, 20 and 30 colonies, each colony has 10 ants. The results are averaged over 10 trials with 3000 and 10000 iterations per trial for kroA100 and lin318 respectively. This based on the assumption that the algorithm ran 3000 iterations for each 100 nodes of the problem instance. The parameter setting are $\beta$=2, $\sigma = \gamma = 0.1$ and q0 = 0.9. The heuristic function used for TSP is the inverse of the distance, i.e., $H_{ij}=1/d_{ij}$.

Figure 2 and Figure 3 show the results of applying ACS, IMACO-AVG and IMACO-MAX on kroA100 and lin318 respectively. For comparison purpose, ACS was run using 10, 20, 30, 40, 50, 70, 90, 120, 150, 200 and 300 ants. Note that since ACS has single ant colony, so in Figure 2 and Figure 3 the no. of ants used by ACS is equal to no. of colonies *10. As shown in these two figures, the results of ACS show that increasing the number of utilized ants in both experiments result in a worsen performance. This means that ACS can not benefit form the increase in the number of utilized ants, the algorithm always get trapped in stagnation situation and can not improve the solution quality. The better results obtained when the number of ants is 20-30.

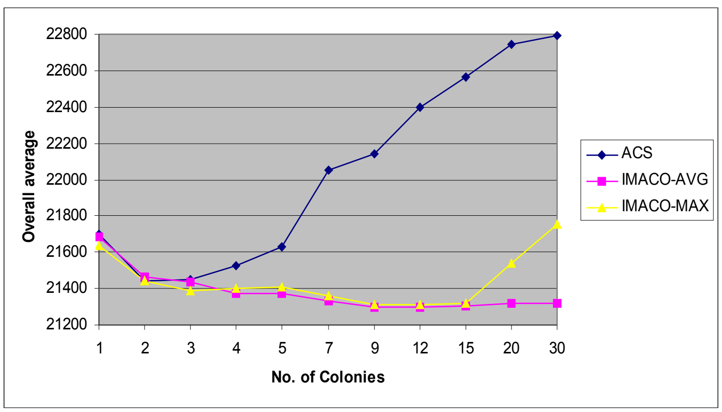


*Figure 2:* KroA100 ACS, IMACO-AVG and IMACO-MAX performance comparison

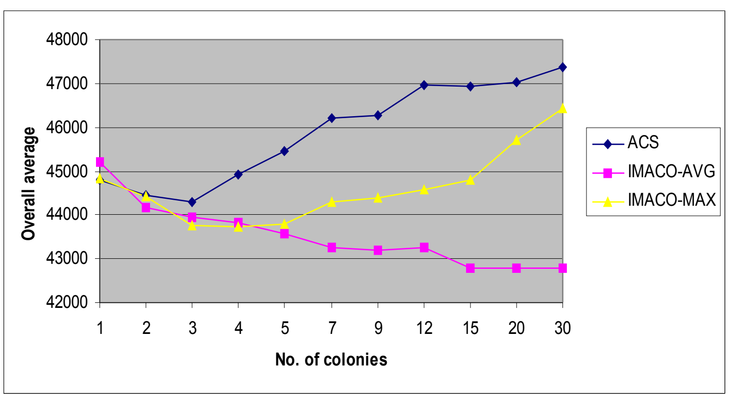


*Figure 3:* Lin318 ACS, IMACO-AVG and IMACO-MAX performance comparison

These results clearly show IMACO-AVG is able to improve its performance by utilizing an increasing the number of ants' colonies. Whereas, the results of IMACO-MAX show that this algorithm get limited success in improving its performance for certain number of utilized colonies. However, this performance started to decline when utilizing more than 15 colonies for kroA100 and more than 5 colonies for lin318.

It is obvious that both algorithms obtained better results in term of the overall average solution for both problem instances. The superior of IMACO-AVG is clear as this algorithm shows a stabilizing performance using the increased number of colonies. The average pheromone evaluation technique was a successful organizing technique of the ants activities up to 30 colonies utilized. The max pheromone evaluation techniques is comparable to the other technique for certain number of colonies but unfortunately the performance get worse as the number of colonies significantly increased. Another experiment ran using IMACO-AVG with 10 colonies on different instances of TSP. The results are reported in Table 1 and compared with those of ACS and MMAS. In this table, the optimal solution for each TSP instance is shown in each column header and all reported results of all algorithms represent the overall average solution. The results of ACS and MMAS are taken from the literature (Dorigo and Gambardella, 1997; Stützle and Hoos, 2000). The number of iteration IMACO-AVG ran on each instance was according to the assumption made by Dorigo and Gambardella (1997) which is equal to *10000*no of nodes/no of ants*.

Table 1: *IMACO-AVG results compared with ACS and MMAS*

| | TSP instance | | | | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **KroA100 Opt: 21282** | **d198 Opt: 15780** | **Lin318 Opt: 42029** | **att532 Opt: 27686** | **rat783 Opt: 8806** | **fl1577 Opt: 22137** |
| **IMACO-AVG** | 21298.5 | 15966.6 | **42341.6** | **28075.7** | **8932.8** | **22520.3** |
| **ACS** | 21420.0 | 16045 | 43296.85 | 28522.8 | 9066.0 | 23136.0 |
| **MMAS** | **21291.6** | **15956.8** | 42346.60 | 28112.6 | 8951.5 | NA |

The results show that IMACO-AVG outperformed ACS for all instances. Comparing IMACO-AVG with MMAS - the best known ant algorithm - for the first two instances MMAS was the best but the results of IMACO was very close. For the other bigger instances where the chance of stagnation increases IMACO-AVG outperformed MMAS giving the best average solution.

## 5.0 CONCLUSION AND FUTURE WORK

A new algorithmic approach has been proposed; it divides the ants' population into number of colonies and effectively coordinates their works. Based on the proposed pheromone evaluation function, two variants of this approach have been developed namely IMACO-AVG and IMACO-MAX. The results show that both variants can outperform the ACS algorithm with similar number of ants. However, the results obtained from IMACO-AVG were better than those obtained from IMACO-MAX. IMACO-AVG has been furthermore tested on different TSP and compared with ACS and MMAS and its superior performance was oblivious especially when applied on big size instances which contain a high risk of stagnation.

Testing new pheromone evaluation mechanisms is a possible future direction. Another interesting future work is testing different values for the parameters involved in the exploration / exploitation control mechanism and investigating the best range for each parameter.

## REFERENCES


Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A new ranked-based version of the ant system: a computational study. *Central European Journal for Operations Research and Economics*, 7(1), 25-38.

Cordon, O., Fernandez, I., Herrera, F., & Moreno, L. (2000). A new ACO model integrating evolutionary computation concepts: The Best-Worst Ant System. In: M. Dorigo, M. Middendorf, and T. Stützle, editors, *Abstract Proceedings of ANTS2000- From Ant Colonies to Artificial Ants: A Series of International Workshops on Ant Algorithms*, 22-29.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics,* Part B*,* 26(1), 29-41.

Dorigo, M., & Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1(1), 53-66.

Dorigo, M., Birattari M. & Stützle T. (2006). Ant Colony optimization: Artificial Ants as Computational Intelligence Technique. *IEEE Computational Intelligence Magazine*, Nov. 2006.

Kawamura, H., Yamamoto, M., Suzuki, K., & Ohuchi, A. (2000). Multiple ant colonies algorithm based on colony level interactions. In *IEICE Trans. Fundamentals*, E83-A(2).

Sim, K. M., & Sun, W. H. (2003). Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Trans. on System, Man, and Cybernetics- Part A: System and Human,* 33(5), 560-572.

Stützle, T., & Hoos, H. (2000). Max-Min Ant System. *Journal* of *Future Generation Computer Systems*, 16, 889 – 914.