ANALYSIS OF THE STAGNATION BEHAVIOR OF THE INTERACTED MULTIPLE ANT COLONIES OPTIMIZATION FRAMEWORK

ALAA ALJANABY
1Computer Science Section, College of Arts and Sciences, University of Nizwa, Oman
alaa@unizwa.edu.om

KU RUHANA KU-MAHAMUD
2School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Malaysia
ruhana@uum.edu.my

ABSTRACT

Search Stagnation is a common problem that all Ant Colony Optimization (ACO) algorithms suffer from regardless of their application domain. The framework of Interacted Multiple Ant Colonies Optimization (IMACO) is a recent proposition. It divides the ants’ population into several colonies and employs certain techniques to organize the work of these colonies. This paper conducts experimental tests to analyze the stagnation behavior of IMACO. It also proposes the idea that different ant colonies use different types of problem dependent heuristics. The performance of IMACO was demonstrated by comparing it with the Ant Colony System (ACS) the best performing ant algorithm. The Computational results show the superiority of IMACO. The results show that IMACO suffers less from stagnation than ACS.

Keywords: Ant colony optimization, combinatorial optimization problems, search stagnation, pheromone evaluation, exploration, exploitation.

1. INTRODUCTION

Several ACO algorithms are presented in the literature among them Ant Colony System (ACS) and Max-Min Ant System (MMAS) are the best performing algorithms [10, 11]. These algorithms can get a good solution at the early stages of the search process but unfortunately all ants quickly converged to a single solution and then the algorithm is unable to improve that solution [7]. 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.

The recently proposed IMACO framework tries to improve the performance of ACO algorithms by utilizing several ant colonies with certain techniques to organize the work of these colonies. The proposed framework composes necessary techniques that encourage the controlled exploration of the search space in couple with a good exploitation of previously obtained good solutions. Exploration allows ants to look at the search space for some new solutions. This should be done under certain control to avoid exploring a very wide area of the search space that might be far from the optimal solution. On the other hand a good exploitation of the search history is necessary to search the solution space in the neighborhood of previously found good solutions. However, a very strong exploitation is not required because it increases the convergence speed of ants towards the same solution obtained in previous iterations [1, 2, 3].

Machine scheduling problems, traveling salesman problem, quadratic assignment problem, vehicle routing problem, and network routing problem are some well known examples of Combinatorial Optimization Problems that have great importance in research and development. These problems have a discrete set of feasible solutions and the goal is to find the optimal solution (the best solution from the feasible solutions). These problems are theoretically proven as NP - hard problems [6]. The only way to tackle these problems is to use approximate (heuristic) algorithms such as tabu search, evolutionary computation, simulated annealing, genetic algorithms and ACO. Single Machine Total Weighted Tardiness Problem (SMTWTP) is an important combinatorial optimization problem that considers the job scheduling for sequential processing on a single machine and the target is to minimize the total...
tardiness of all jobs. This is a crucial task in manufacturing and production planning. SMTWTP is used as the application domain of this work.

The rest of this paper is organized as follows. The framework of IMACO is described in section 2. A brief description of SMTWTP is given in Section 3. In section 4, an experimental study is conducted to analyze the stagnation behavior of IMACO and compare it with that of ACS. The result of applying IMACO with different problem dependent heuristics to solve all 375 available instances of SMTWTP is also presented. The final section concludes the paper.

2. INTERACTED MULTIPLE ANT COLONIES OPTIMIZATION

IMACO framework is recently proposed in previous work of the author [1, 2, 3]. In this framework there are two levels of interaction the first one is the colony level and the second one is the population level. 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 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:

\[
j = \begin{cases} \arg \max_{j \in N_i^k} \{ f(P_n) H^{\beta}_{ij} \} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases}
\]

(1)

The random variable S is selected according to the following probabilistic rule:

\[
S = \begin{cases} \frac{f(P_n) H^{\beta}_{ij}}{\sum_{j \in N_i^k} f(P_n) H^{\beta}_{ij}} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases}
\]

(2)

Where \( N_i^k \) is the set of remaining nodes to be visited by the \( k^{th} \) ant of colony \( v \) located at node \( i \), \( P_n \) is the pheromone of colony \( v \) on the edge \((i, j)\) and \( f(P_n) \) is the evaluation function of the pheromone on the edge \((i, j)\) that will be discussed in Section 3.

Global and local pheromone updating are used in IMACO. 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 (global) solution since the starting of the algorithm execution or the ant that got the best solution in the current iteration of the algorithm execution. In this work a combination of so far best and iteration best ants are allowed to update the pheromone.

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 is allowed to deposit an amount of the colony’s pheromone on the edges of its tour according to the following global pheromone update:

\[
P_{ij}^v = (1 - \sigma) P_{ij}^v + \sigma \Delta P_{ij}^{v,bs}
\]

(3)

Where \( \sigma \) is a pheromone evaporation parameter its value is in the range \([0, 1]\) and \( \Delta P_{ij}^{v,bs} \) is the pheromone quantity added to the connection \((i, j)\) belonging to the best solution of the \( v^{th} \) colony \( L^{v,bs} \) and is given by:

\[
\Delta P_{ij}^{v,bs} = \begin{cases} 1 / L^{v,bs} & \text{if } (i, j) \text{ belongs to the solution of colony } v \\ 0 & \text{otherwise} \end{cases}
\]

(4)

To create a search diversification IMACO uses iteration best solution once in the pheromone updating after each 50 times of using the global best solution [8, 9]. Local pheromone updating includes that each ant 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. Local pheromone update is applied by each ant on the visited edges. 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:

\[
P_{ij}^v = (1 - \gamma) P_{ij}^v + \gamma P_0
\]

(5)
Where \( P_0 \) is the initial pheromone value and \( \gamma \) is another pheromone evaporation parameter with a value in the range \([0, 1]\).

### 2.1 PHEROMONE EVALUATION TECHNIQUE

The pheromone of different colonies has been evaluated using two mechanisms. The first mechanism evaluates 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 average pheromone evaluation function \( f(P_0) \) on the edge \((i, j)\) for IMACO-AVG will be defined as:

\[
f(P_0) = \frac{\sum_{s=1}^{M} P_{ij}^s}{M}
\]  

(6)

The second mechanism evaluates the pheromone as the maximum value of the pheromone values of all colonies on some edge. This mechanism, referred as IMACO-MAX, chooses the max value among the available \( M \) values. The pheromone evaluation function for IMACO-MAX is defined as:

\[
f_2(P_0) = \text{Max}_{\forall s} P_{ij}^s
\]  

(7)

The above rule lets an ant’s decision to choose some new edge be based on the best available experience of ants of all colonies that previously visited this edge. This kind of cooperation using max pheromone evaluation is trying to make an early exploitation of the history of the search by choosing the max (best) available pheromone value. The result of this max pheromone evaluation function lets an ant to follow the best available information about the goodness of particular edge. However, since best pheromone comes from different colonies, this will provide necessary diversification that helps ants’ to avoid the attraction to a one good solution.

The above two mechanism are pure average and max evaluation that depends 100% on the average evaluation function. The following rule is a more general which evaluates the pheromone as a composition between the pheromone values of the ant own colony and the value of the pheromone evaluation function based on some pheromone evaluation rate. Consider that the composition rate is 0.5; an ant will build 50% of its decision based on its own colony’s experience and the other 50% based on the experiences of other colonies. This new variant will be called IMACO-AVG \( E \lambda \) and IMACO-MAX \( E \lambda \) where \( \lambda \) is the pheromone evaluation rate; its value is in the range \([0, 1]\). The pheromone evaluation function is then defined as:

\[
f'((P_0)) = \lambda P_{ij}^s + (1 - \lambda) f((P_0))
\]  

(8)

Where \( P_{ij}^s \) is the pheromone belongs to colony \( s \) on edge \((i, j)\). Note that IMACO-AVG \( E0 \) and IMACO-MAX \( E0 \) represent the pure pheromone evaluation and IMACO-AVG and \( E1 \) IMACO-AVG represent no interaction between utilized ant colonies.

### 2.2 EXPLORATION TECHNIQUE

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. Exploration and exploitation is controlled by the parameter \( q_0 \) whose value is in \([0, 1]\). It 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). Setting \( q_0 \) to zero means that the algorithm uses a pure exploration while pure exploitation is reached by setting \( q_0 \) to one. However, the value used for \( q_0 \) in many research papers usually between 0.5 and 0.9 \([1, 2]\). Most of the work done using ACS in solving different problems was with \( q_0 = 0.9 \) which gives the algorithm a high chance of exploitation without losing the chance of exploration.

IMACO considers the case where different ants’ colonies have different values for the parameter \( q_0 \). The value 0.8 has been assigned to the centre colony whose number equal to int (no. of colonies / 2). This value is increased / decreased for the colonies after / before the centre colony by a changing factor called QCF. This technique enables the utilized ant colonies to work with different levels of exploration. Some will prefer high exploration of new areas of search space while other colonies will prefer high exploitation search history.
3. THE SINGLE MACHINE PROBLEM

SMTWTP can be stated as follows. Each of n jobs is to be processed without pre-emption on a single machine that can handle no more than one job at a time. The processing and set-up requirement of any job are independent of its position in the sequence. The release time of all jobs is zero. Thus, jobs \( j (j=1, \ldots, n) \) becomes available at time zero, requires uninterrupted positive processing time \( p_j \), which includes set-up and knock-down times on the machine, has a positive weight \( w_j \), and has a due time \( d_j \) by which it should ideally be finished. For a given processing order of the jobs, the completion time \( c_j \) and the tardiness \( T_j = \max\{0, c_j - d_j\} \) of job \( j \) can be computed. The problem is to find a processing order of the jobs with minimum total weighted tardiness \( \sum_{j=1}^{n} W_j T_j \) [4, 5].

The SMTWTP is an NP-hard scheduling problem for which instances with more than 50 jobs often cannot be solved to optimality with state of the art branch and bound algorithms [8]. The total number of available instances is 125 for values of \( n=40, n=50 \) and \( n=100 \). Optimal values of solutions are available for 124 and 115 of 40 and 50 job problem instances respectively. The values for unsolved problems are the best known solution to [8]. These solutions appear to be optimal since they have not been enhanced for a long time. The best known solutions to date of the 100-job instances are available and most of them are according to [8, 9].

Three types of problem specific heuristic are examined in this work. These problem specific heuristic are easily calculated and have been studied in the literature [8] and are as follows.

- **Earliest Due Date (EDD):** this heuristic puts the jobs in non-decreasing order of the due dates \( d_j \) and given by:
  \[
  H_y = \frac{1}{d_j} \quad (9)
  \]

- **Modified Due Date (MDD):** this heuristic puts the jobs in non-decreasing order of the modified due dates \( mdd \), which given by \( mdd = \max\{C + p_j, d_j\} \), where \( C \) is the sum of the processing times of the already scheduled jobs. This heuristic is given by:
  \[
  H_y = \frac{1}{mdd_j} \quad (10)
  \]

- **Apparent Urgency (AU):** this heuristic puts the jobs in non-decreasing order of the apparent urgency which given by:
  \[
  au_j = \frac{w_j}{p_j} \exp\{-\frac{\max\{d_j - C_j, 0\}}{k \bar{P}}\} \quad (11)
  \]

  Where \( \bar{P} \) is the average processing time of the remaining jobs, \( k \) is a scaling parameter which set to 2 [8]. The heuristic is given by:
  \[
  H_y = \frac{1}{au_j} \quad (12)
  \]

4. SEARCH STAGNATION ANALYSIS

ACS, IMACO-AVG and IMACO-MAX for SMTWTP have been implemented using visual C++. Both versions of IMACO have been applied to all available 375 instances of SMTWTP. Based on past work of the authors the number of colonies utilized by IMACO-AVG and IMACO-MAX was 8 colonies, the evaluation rate was \( \lambda = 0.4 \) and the exploration / exploitation control parameter was \( \text{QCF} = 0.025 \) [2, 3]. In addition to use IMACO with EDD, MDD and AU, this section is developing the idea of using IMACO with different combinations of the three heuristics. For instance, using EDD-MDD means that half of the utilized ant colonies will use EDD while the other half of these colonies will use MDD.

The global pheromone updating is performed by according to rules 3 and 4. The value of best solution (global-best or iteration-best) mentioned in rule 4 represents the total weighted tardiness of the jobs sequence of the best solution. Local pheromone updating is performed using rule 5 and \( P_0 \) the initial value of pheromone trials that usually assigned a small value computed as \( P_v = \frac{1}{n T_{EDD}} \) where \( n \) is the number of jobs and \( T_{EDD} \) is the total weighted tardiness of job sequence obtained by EDD.

The stagnation behavior of IAMCO is the main concern of this paper. This section tries to experimentally analyze the stagnation behavior by observing 100 typical runs of ACS, IMACO-MAX, and IMACO-AVG and calculate how many times that each algorithm get trapped into stagnation situation. Figure 1 shows the results of tracing one typical run of the three algorithms on the 1st SMTWTP 100-job. ACS ran with 10 ants, IMACO-AVG E.4 and IMACO-AVG E.4 with 8 colonies of 10 ants each, all algorithms ran with MDD.
The results in Table 1 show the so far best solution obtained after completing certain number of iterations. It obvious that ACS can not improve its solution after 70% of algorithm iterations completed, the last 30% of the algorithm iterations ACS search process was stagnant. IMACO-AVG was the best among the other algorithms. It was able to direct its search away from stagnation situation and improve its solution.

Table 2 shows the results of experiments done on 125 instances of 40, 50 and 100 job SMTWTP. The results presented in these tables are the number the optimal solution found (out of 125). The results of ACS presented in Table 2 are of the implementation developed with this research work. The reason is that the results of ACS presented in the literature usually with local search while all results presented here are without using local search. It is important to mention that all algorithms ran exactly the same number of computation steps.

Table 1. Stagnation situations in 100 trials typical run on 100-job SMTWTP instance

<table>
<thead>
<tr>
<th>Iterations completed</th>
<th>ACS</th>
<th>IMACO-MAX</th>
<th>IMACO-AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40%</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50%</td>
<td>10</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>60%</td>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>70%</td>
<td>20</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>80%</td>
<td>24</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>90%</td>
<td>28</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>100%</td>
<td>28</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 1:** One trial typical run on 100-job SMTWTP instance

Table 1 shows a result of 100 of typical runs like those described above. The numbers in the table shows how many times the algorithm cannot improve its solution from the previous step. The algorithm which suffers more from stagnation was ACS. The chance of stagnation increases after 50% of iterations completed because all ants converged to one solution or one dominant path with highest amount of pheromone and ability to get out from this situation decrease with the time. IMACO-AVG was the algorithm that suffers less from stagnation. This proves that the techniques incorporated in this algorithm did well to avoid such situations.

5. CONCLUSION

It is obvious based on the above results that IMACO-AVG and IMACO-MAX outperform ACS in terms of the number of optimal solutions found. IMACO-AVG was the best algorithm that found the best results all the
way. It is the ability of IMACO to avoid the stagnation situation and improves its solutions with the time. This comes from the kind of interaction used between ant colonies and the type of information used by ants when making their decision. The proposed interaction plays on two directions which are cooperation and diversification. Pheromone evaluation mechanism plays the main role in cooperation. Pheromone evaluation was the mean to combine the pre-acquired information about the quality of the solutions represented as pheromone values.

Average pheromone evaluation was the best technique that puts IMACO-AVG in front of other state-of-the-art ant algorithms. Pheromone evaluation needs a high support from other mechanisms. On the other hand, letting different colonies works with different levels of exploration / exploitation was of great aid in achieving diversification. Some colonies prefer a higher exploration while others prefer a higher exploitation. This provides the whole search process with a wide range of good solution that ants of different colonies choose their best solution from.

References