ANT COLONY OPTIMIZATION FOR MANUFACTURING RESOURCE SCHEDULING PROBLEM

Wang Su, Meng Bo
Computer School, Wuhan University, Hubei Province, China; Email: zzwangsu@163.com.

Abstract: Effective scheduling has a significant influence on efficiency of resources and manufacturing systems. In this paper, we propose a model of resource scheduling based on matching mechanism and apply Ant Colony Optimization to solve this problem. During the process of seeking, ants can change the size of colony adaptively and incline to select the resources which cost less. Finally, we examine the performance of optimization in a test case scenario in container terminals and obtain satisfactory results that suggest that ACO is well suited for application in practice.

Key words: Ant Colony Optimization, Manufacturing Resource Scheduling, Pheromone.

1. INTRODUCTION

Information Technology not only creates new opportunities for manufacturing industry but also gives rise to new challenges and changes. The most prominent is the requirement for higher productivity in the whole manufacturing process. To deal with the challenge, an effective scheduling system that can improve its performance is essential. Scheduling is no longer a static optimization problem. Most practical scheduling problems are NP-hard, dynamic and unpredictable. These manufacturing characteristics limit the effectiveness of conventional scheduling approaches. At present, a key factor of successful scheduling is heuristic optimization.

There are many efforts in the field of optimization of manufacturing resource scheduling. S.G.Ponnambalam adopted Tabu Search (TS) to resolve Flexible Scheduling Problem. Gehlsen proposed a framework for discrete event simulation and applied Genetic Algorithm (GA) to get optima.

This project is supported by the Natural Science Foundation of Hubei Province of China under the grant No.2001ABB058 and the Post Doctoral Science Foundation of China.

Please use the following format when citing this chapter:

In this paper, we introduce a general scheduling model based on matching mechanism. Consider that ACO Algorithm is a good meta-heuristic approach to tackle hard combinatorial optimization problems and it is parallel, robust and positive feedback, we adopt ACO Algorithm to resolve this kind of problem.

2. MANUFACTURING SCHEDULING PROBLEM BASED ON MATCHING

The manufacturing resource scheduling problem based on matching mechanism is described as follows: Resources are scheduled by multiple tasks. Each task can schedule more resources, but one resource can only be scheduled by one task. Resources have different costs. The goal is to determine a matching solution between tasks and resources so that the total cost is minimized on the premise of all tasks are completed. Tasks are defined as \((T_1, T_2, T_3, ..., T_n)\). Resources are defined as \((R_1, R_2, R_3, ..., R_m)\). \(t_i\) is the need of task \(T_i\), \(c_j\) is the cost of scheduling resource \(R_j\).

Minimize: \(\min \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} c_j\)

Subject to: \(\sum_{j=1}^{m} x_{ij} c_j \geq t_i\)

\[
x_{ij} = \begin{cases} 
1 & \text{if } T_i \text{ schedules } R_j \\ 
0 & \text{otherwise} 
\end{cases}
\]

\(\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \leq m\)

3. ANT COLONY OPTIMIZATION

ACO is an evolutionary approach introduced by Dorigo. In ACO several generations of artificial ants search for good solutions. Every ant of a generation builds up a solution step by step. Ants that find a good solution mark their paths through the decision space by putting some amount of pheromone on the path. The following ants of the next generation are attracted by the pheromone so that they will search in the solution space near good solutions.
For scheduling problem in this paper, each ant is assigned to one task and each note represents one resource. $T_i$ schedules resources, then ant $k_i$ deposits pheromone on the path. $\tau_{ij}(t)$ is the pheromone of path $l_{ij}$ which represents the cost of scheduling $R_j$ after $R_i$. For example, there are 10 available resources ($n_1, n_2, n_3, ..., n_{10}$) in Figure 1. All ants set out from the initial note $n_0$ at the beginning. Ants establish four resource unions $\{R_1, R_2, R_4\}$, $\{R_2, R_8, R_{10}\}$, $\{R_6, R_9\}$, $\{R_7\}$ for four tasks and $R_5$ is not scheduled. The union $\{R_1, R_3, R_4\}$ schedules $R_1$, $R_3$ and $R_4$, and ants deposits pheromone on path $l_{01}$, $l_{13}$, $l_{34}$.

![Figure 1. Ten resource scheduled by four tasks](image)

The node transition rule is as follows. Ant $k$ in $R_i$ chooses $R_j$ following the rule:

$$
S = \begin{cases} 
\arg \max_{j \in \text{tabu}} \left\{ \tau_{ij}^\alpha \cdot \eta_{ij}^\beta \right\} & q \leq q_0 \\
S & q > q_0 
\end{cases}
$$

(1)

$q$ is a random variable between $[0,1]$, $q_0$ is a tuneable parameter between $[0,1]$. For the selection of a resource the ant uses pheromone information denoted by $\tau_{ij}$ and heuristic information denoted by $\eta_{ij}$. The heuristic value is generated by some problem such as resource ability. With probability $q_0$, the ant chooses $R_j$ from the set of resources that have not been scheduled so far which maximizes $\tau_{ij}^\alpha \cdot \eta_{ij}^\beta$, where $\alpha$ and $\beta$ are constants that determine the relative influence of pheromone values and heuristic values. With the probability $1 - q_0$ the next resource is chosen according to $S$ determined by $p_{ij}^k$.

$$
p_{ij}^k = \begin{cases} 
\frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{u \in \text{tabu}} \tau_{iu}^\alpha \cdot \eta_{iu}^\beta} & j \in \text{allow } k \\
0 & \text{otherwise}
\end{cases}
$$

(2)
Unlike Traveling Salesman Problem (TSP), ants here seek route for forming some resource-unions for tasks. The ant will stop seeking when it arrives at a certain resource note and finds that the current resource union can accomplish task $T_i$. It’s unnecessary to visit the other resource note. So every time the ant arrives at a resource note, it needs to accumulate the current resource ability and judge whether the combinational ability ($ra_i$) meets the task demand ($t_i$). If $ra_i$ is higher than $t_i$, the ant $k$, will stop seeking, otherwise continue. When all the tasks complete, all ants stop and one circle finishes. If not all of the tasks can be accomplished after ants visits all resource notes, the system will make some tasks wait.

In ACO, all ants are allowed to deposit pheromone after completing their tours. The updating rule is as follows:

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} + (1 - \rho) \Delta \tau_{ij} \quad (3)$$

$$\Delta \tau_{ij} = \begin{cases} \frac{t_k}{Q_k} & \text{ant } k \text{ select note } i \text{ and note } j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$(i,j)$ is one path of the best tour, $\rho$ is a parameter governing pheromone decay. The reason for this is that old pheromone should not have a too strong influence on the future. $Q_k$ is the real cost of completing $T_k$, which is always higher than $t_k$.

The algorithm stops when some stopping criterion is met, e.g. a certain number of generations have been done or the best found solution has not changed for several generations.

Using ACO Algorithm to solve resource scheduling problem, the pseudocode is as follows:

Step1: Initialization.

Step2: while (available resource) and (uncompleted task)

for (k=1;k<=n;k++)

if (task k has completed) Continue;
else Ants select next paths according to formulation (1), (2);

Step3: All ants complete their searching process;

if (total cost of current solution < total cost of best solution)

Update pheromone according to formulation (3), (4);

Step4: NC++; 

if (NC < NC_{max}) or (no evolving solution) Goto Step2;
else Output the solution with minimal cost;
End the whole program.
4. SIMULATION EXPERIMENTS

In this section we describe a test case scenario of scheduling problem in container terminals using ACO Algorithm. Faced with the increasing growth of container throughput and more large ships, the demands on ports of upload/load task with maximum efficiency become more important. Tugboat scheduling is one important scheduling in the whole task.

When ships arrive at a port, they can not enter into the berth directly and need to be tugged by the tugboats. Moreover, the moving between two berths and the department of vessels also need to be tugged. Vessels with different sizes and types should schedule tugboats with different horsepower, avoiding big tugboat for small vessel and small tugboat for big vessel. On the other hand, one tugboat is only scheduled by one vessel simultaneously. To make sure scheduling can be done as rapidly as possible with least cost; tugboats necessary for this procedure must be scheduled at an optimum level.

We make simulation experiments of tugboat scheduling. Table 1 shows the allocation of tugboat at some Chinese ports. Table 2 shows some records of arriving vessel we use in this simulation.

<table>
<thead>
<tr>
<th>Horsepower</th>
<th>1200ps</th>
<th>2600ps</th>
<th>3200ps</th>
<th>3400ps</th>
<th>4000ps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port 1</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Port 2</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Port 3</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Port 4</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1: Allocation of Tugboat

<table>
<thead>
<tr>
<th>Name</th>
<th>Country</th>
<th>Time</th>
<th>Length(m)</th>
<th>Width(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YuHe</td>
<td>China</td>
<td>03-05-20 10:00</td>
<td>200</td>
<td>28</td>
</tr>
<tr>
<td>Aoleta</td>
<td>Poland</td>
<td>03-05-20 10:00</td>
<td>229</td>
<td>32</td>
</tr>
<tr>
<td>BuleStar</td>
<td>Britain</td>
<td>03-05-20 10:00</td>
<td>155</td>
<td>25</td>
</tr>
<tr>
<td>Wasalian</td>
<td>Singapore</td>
<td>03-05-22 14:20</td>
<td>67</td>
<td>14</td>
</tr>
<tr>
<td>Ankejian</td>
<td>Singapore</td>
<td>03-05-21 14:00</td>
<td>135</td>
<td>31</td>
</tr>
<tr>
<td>JiaoCheng</td>
<td>China</td>
<td>03-05-21 14:00</td>
<td>148</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 2: Records of Arriving Vessels

The results of simulation without ACO Algorithm are described in Hu Wenbin\textsuperscript{7}. We use ACO to optimize simulation results. In the real discreet scheduling problem, the number of ants is a variable parameter which can be changed according to the number of tasks. Other parameters are: $\tau_0 = 0.5$, $\alpha = 1$, $\beta = 3$, $\rho = 0.8$, $NC_{\text{max}} = 5000$. The results are the average of ten
experiments and showed in Table 3: (1) is tugboat utilization rate, (2) is average waiting time of vessel (minute), (3) is max waiting queue length.

<table>
<thead>
<tr>
<th>Method</th>
<th>Simulation Without ACO</th>
<th>Simulation With ACO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Evaluation</td>
<td>(1)</td>
</tr>
<tr>
<td>Port 1</td>
<td></td>
<td>23.6%</td>
</tr>
<tr>
<td>Port 2</td>
<td></td>
<td>20.5%</td>
</tr>
<tr>
<td>Port 3</td>
<td></td>
<td>18.9%</td>
</tr>
<tr>
<td>Port 4</td>
<td></td>
<td>19.5%</td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this paper we propose a general model for scheduling based on match mechanism and adopt ACO Algorithm to solve this problem.

In the algorithm, ants can change the size of colony adaptively and incline to select the resources which cost less. The results of simulation show that compared with only simulation, applying ACO to optimize the scheduling process can improve the utilization rate of resources and the efficiency of tasks.

Our feature work is to research ACO Algorithm deeply to improve the convergence speed and avoid local optima.

6. REFERENCES