Ant Colony Optimization Based on Adaptive Volatility Rate of Pheromone Trail

Zhaoquan CAI1, Han HUANG2,3, Yong QIN4, Xianheng MA2
1Educational Technology Center, Huizhou University, Huizhou, China
2School of Software Engineering, South China University of Technology, Guangzhou, China
3State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China
4Center of Information and Network, Maoming University, Maoming, China
Email: {gdzqcai, bssthh}@163.com
Received May 25, 2009; revised July 15, 2009; accepted August 13, 2009

Abstract

Ant colony optimization (ACO) has been proved to be one of the best performing algorithms for NP-hard problems as TSP. The volatility rate of pheromone trail is one of the main parameters in ACO algorithms. It is usually set experimentally in the literatures for the application of ACO. The present paper first proposes an adaptive strategy for the volatility rate of pheromone trail according to the quality of the solutions found by artificial ants. Second, the strategy is combined with the setting of other parameters to form a new ACO method. Then, the proposed algorithm can be proved to converge to the global optimal solution. Finally, the experimental results of computing traveling salesman problems and film-copy deliverer problems also indicate that the proposed ACO approach is more effective than other ant methods and non-ant methods.

Keywords: Ant Colony Optimization (ACO), Adaptive Volatility Rate, Pheromone Trail

1. Introduction

ACO was first proposed by M. Dorigo and his colleagues as a multi-agent approach to deal with difficult combinatorial optimization problems such as TSP [1]. Since then, a number of applications to the NP-hard problems have shown the effectiveness of ACO [1]. Up till now, Ant Colony System (ACS) [2] and MAX-MIN Ant System (MMAS) [3] are so successful and classical that their strategies such as pheromone global-local update and Maximum-Minimum of pheromone are widely used in recent research [1].

The main parameters of ACO may conclude: k, ρ, α and β, where k is the number of artificial ants used for solution construction, ρ is the parameter for volatility of pheromone trail and α,β determines the relative importance of pheromone value and heuristic information [2,4,5]. All of the parameters are usually set with experimental methods in the application of ACO [5–7]. For the adaptive parameter setting, M. Dorigo and L.M. Gambardella presented a formula for the optimal number of ants k based on the value of ρ and q0 in ant colony system. I. Watanabe and S. Matsui proposed an adaptive control mechanism of the parameter candidate sets based on the pheromone concentrations [8]. M. L., Pilat, and T. White put forward the ACSGA-TSP algorithm [9] with an adaptive evolutionary parameters β, ρ, q0 and gave the experimental values of these parameters for some TSP problems. For the parameters α and β, which regulate the relative importance of pheromone trail and closeness [10], H. Huang proposed a dynamic strategy for a bi-directional searching ant colony system [11]. However, other parameters should be set experimentally.

This paper presents a trial work of setting the parameters of ACO adaptively. First, a tuning rule for ρ is designed based on the quality of the solution constructed by artificial ants. Then, we introduce the adaptive ρ to form a new ACO algorithm, which is tested to compute several benchmark instances of traveling salesman problem and film-copy deliverer problem. Finally, the experimental result indicates that the new ACS with adaptive ρ performs better than GA [12], ACO [13] and ACS [2,14]. Furthermore, the convergence of the proposed ACO algorithm is proved.
2. Adaptive Volatility Rate of Pheromone Trail

The framework of ACO [1–2] is inspired by the ants’ foraging behavior in selecting the shortest path between the nest and the food. Each ant builds a tour (i.e. a feasible solution to the TSP) by repeatedly applying a stochastic greedy rule (the state transition rule) as Equation (1) shows:

\[ P_{gr}(t) = \begin{cases} \frac{[\tau_{gr}(t)]^{\alpha}[\eta_{gr}]^{\beta}}{\sum_{r \in J_s(g)}[\tau_{gr}(t)]^{\alpha}[\eta_{gr}]^{\beta}} & \text{if } s \in J_s(g) \\ 0 & \text{otherwise} \end{cases} \]  

(1)

where \( P_{gr} \) is the probability with which the ant \( m \) chooses to move from city \( g \) to city \( s \) in iteration \( t \), \( \tau \) is the pheromone, \( \eta = 1/d \) is the reciprocal of distance \( d_{gr} \), and \( J_s(g) \) is the set of cities not having been visited yet when ant \( m \) is at city \( g \).

After constructing its tour, an artificial ant also modifies the amount of pheromone on the visited edges by applying the pheromone updating rule. The rule is designed so that it tends to give more pheromone to the edges which should be visited by ants. The classical pheromone updating rule is:

\[ \tau_{gr}(t + 1) = (1 - \rho)\tau_{gr}(t) + \rho \Delta \tau_{gr}(t) \]  

(2)

where \( \Delta \tau_{gr}(t) \) is the increment for the pheromone of edge \((g,s)\) at the \( t \)-th iteration, and \( \rho \) is the volatility rate of the pheromone trail. The optimal \( \rho \) was set \( \rho = 0.1 \) experimentally [1,2,4], which means that 90 percent of the original pheromone trail remains and its 10 percent is replaced by the increment.

In order to update the pheromone according to the quality of solutions found by ants, an adaptive rule for volatility of the pheromone trail is designed as follows:

\[ \rho_{ml} = L_{ml}^{-1} / (L_{ml}^{-1} + L_{p}^{-1}) \]  

(3)

where \( L_m \) is the length of the solution \( S_m \) found by ant \( m \), and \( L_p \) is the length of the solution \( S_p \) built based on the pheromone matrix, shown as Equation (4).

\[ s = \arg \max \{ [q(r,u)] \} \]  

(4)

where \( s \) is the city selected as the next one to city \( r \) for any \((r,s) \in S_p\).

The motivation of the proposed rule is: better solutions should contribute more pheromone, and the worse ones contribute less. We will use this rule to design a new ACO algorithm in the following section.

3. An ACO Algorithm with the Adaptive Parameter

In this section, a new ACO algorithm with the adaptive rule (shown as Equation 3) is introduced as follows:

**Algorithm new ACO**

**input**: An instance of TSP or FDP problems

Initialize solutions and pheromone value.

\( S_{best} \leftarrow \text{NULL} \).

**while** termination conditions not met **do**

Construct \( S_p \).

**for** \( i = 1 \) to \( k \) **do** \{ \( k \) is the number of artificial ants \}

\( S_i \leftarrow \text{ConstructSolution}(t) \).

\( \rho_i \) is calculated based on \( S_i \).

if \( ( \text{Length}(S_i) < \text{Length}(S_{best}) ) \) or \( (S_{best} = \text{NULL} ) \) then

\( S_{best} \leftarrow S_i \)

**End**

**End**

**Output**: \( S_{best} \).

**End_algorithm**

The framework of the proposed algorithm is similar to ant colony system (ACS) [2], so are the initialization, solution construction and setting of the parameters \( q_o = 0.9, \ k = 10, \ \alpha = 1 \) and \( \beta = 2 \). There is only an updating rule in the algorithm shown as Equation 5 and 6.

\[ \tau_{gr}(t + 1) = (1 - \rho)\tau_{gr}(t) + \rho L_{l}^{-1} \]  

(5)

where \( \forall (g,s) \in S_i \) and \( \rho_i = L_{l}^{-1} / (L_{l}^{-1} + L_{p}^{-1}) \) for the \( t \)-th iteration.

\[ \tau_{gr}(t + 1) = (1 - \rho)\tau_{gr}(t) + \rho L_{best}^{-1} \]  

(6)

where \( \forall (g,s) \in S_{best} \) and \( \rho_{best} = L_{best}^{-1} / (L_{best}^{-1} + L_{p}^{-1}) \) for the \( t \)-th iteration.

4. Convergence of the Proposed Algorithm

In this section, we give the convergence proof of the new ACO algorithm.

Given an arbitrary path \((g,s)\),

\[ \tau_{gr}(t) \leq (1 - \rho)\tau_{gr}(t) + \rho U \leq (1 - \rho)^{y} \tau_{gr}(t) + \frac{1 - (1 - \rho)^{y}}{1 - (1 - \rho)} \rho U \]  

(7)
where \(0 < t' \leq t\), \(\rho_1 = L^{-1}_{\min} / (L^{-1}_{\min} + L^{-1}_{\max})\), \(U = \max \{\tau_{\tau}(0), (L^{-1}_{\min})^{-1}\}\), \(L_{\min}\) is the length of the worst tour and \(L_{\max}\) is the length of optimal tours.

\[
\tau_{\tau}(t) \geq (1 - \rho_2)\tau_{\tau}(t) + \rho_2 D \leq (1 - \rho_2)' \tau_{\tau}(0) + \frac{1 - (1 - \rho_2)'}{1 - (1 - \rho_1)} \rho_2 D
\]

where \(0 < t' \leq t\), \(\rho_2 = L^{-1}_{\max} / (L^{-1}_{\max} + L^{-1}_{\min})\), \(D = \min \{\tau_{\tau}(0), (L^{-1}_{\max})^{-1}\}\).

Because \(0 < \rho_1, \rho_2 < 1\), \(D \leq \tau_{\tau}(t) \leq U\) when \(t \to \infty\).

\[
P_{ab}(\eta_0) \geq p_0 \sum_{k \in I_{(a)}} \tau_{\eta_0}(t_k) \eta_{\eta_0}^\beta \geq p_0 \left[ \tau_{\eta_0}^{\min}(t_0) \cdot \eta_{\eta_0}^{\beta} \right] \sum_{j \in I_{(a)}} P_{ab}[\eta_0^\beta] \geq p_0 \frac{P_{low} \cdot \eta_{\eta_0}^{\beta}}{k \eta_{\eta_0}^{\beta} \max(\eta_0)}
\]

where \(p_0 = P(q > q_0)\) [2], \(\eta_{\eta_0} = \min \{\eta_0\}\) and \(\eta_{\eta_0} = \max \{\eta_0\}\).

Given \(a_0 = p_0 \frac{P_{low} \cdot \eta_{\eta_0}^{\beta}}{k \eta_{\eta_0}^{\beta} \max(\eta_0)} < 1\), the probability, by which \(S^*\) can be found by ants in iteration \(t_0\), is 

\[
P_{S^*}(t_0) = \prod_{(a,b) \in S^*} P_{ab}(\eta_0) \geq a_0^{n-1}, \text{ where } n \text{ is the number of cities.}
\]

The probability, by which \(S^*\) can never be found from iteration \(t_0\), is:

\[
P_{S^*}(t_0) = \prod_{l=t_0}^{\infty} \left[ 1 - \prod_{(a,b) \in S^*} P_{ab}(t_0) \right]^k
\]

where \(k\) is the number of artificial ants and \(t_0\) can be arbitrary.

Hence, \(S^*\) can be found by probability one when the iteration \(t \to \infty\), which theoretically confirms the capacity of global optimization of the proposed ACO algorithm.

5. Numerical Results

This section indicates the numerical results in the experiment that the proposed ACO algorithm is used to solve TSP problems [15] and FDP problems [14]. Other approaches for the problems ACS [2], ACO [13], GA-FDP [12] and ACS-FDP [14] are also tested in the same machines as the comparison with the proposed ACO.

Several TSP instances are computed by ACS [2], ACO [13] and the proposed ACO on a PC with an Intel Pentium 550MBHz Processor and 256MB SDR Memory, and the results are shown in Table 1. It should be noted that every instance is computed 20 times. The algorithms are both programmed in Visual C++6.0 for Windows System. They would not stop until a better solution could be found in 500 iterations, which is considered as a virtual convergence of the algorithms.

Table 1 shows that the proposed ACO algorithm (PACO) performs better than ACS [2] and ACO [13]. The shortest lengths and the average lengths obtained by PACO are shorter than those found by ACS and ACO in all of the TSP instances. Furthermore, it can be concluded that the standard deviations of the tour lengths obtained by PACO are smaller than those of another algorithms. Therefore, we can conclude that PACO is proved to be more effective and steady than ACS [2] and ACO [13]. Computation time cost of PACO is not less than ACS and ACO in all of the instances because it needs to compute the value of volatility rate \(k+1\) times per iteration. Although all optimal tours of TSP problems cannot be found by the tested algorithms, all of the errors for PACO are much less than that for another two ACO approaches. The algorithms may make improvement in solving TSP when reinforcing heuristic strategies like local search like ACS-3opt [2] and MMAS+rs [3] are used.

FDP problem is an extended style of TSP problem. Two FDP instances in the literature [14] are computed by GA-FDP [12], ACS-FDP [14] and the proposed ACO-FDP on a PC with an Intel Pentium 400MBHz Processor and 128 MB EMS memory, and the results are shown in Table 2. It should be noted that every instance is computed 20 times. The algorithms are both programmed in Visual C++6.0 for Windows System. They would not stop until a better solution could be found in 500 iterations, which is considered as a virtual convergence of the algorithms.
Table 1. Comparison of the results obtained by ACS [2], ACO [3] and the proposed ACO (PACO) in TSP instances.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>best</th>
<th>ave</th>
<th>time(s)</th>
<th>standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>kroA100</td>
<td>ACS</td>
<td>21958</td>
<td>22088.8</td>
<td>65</td>
<td>1142.77</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>21863</td>
<td>22082.5</td>
<td>94.6</td>
<td>1265.30</td>
</tr>
<tr>
<td></td>
<td>PACO</td>
<td>21682</td>
<td>22076.2</td>
<td>117.2</td>
<td>549.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ts225</td>
<td>ACS</td>
<td>130577</td>
<td>1313195</td>
<td>430.6</td>
<td>7038.30</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>130568</td>
<td>132984</td>
<td>439.3</td>
<td>7652.80</td>
</tr>
<tr>
<td></td>
<td>PACO</td>
<td>130507</td>
<td>131560</td>
<td>419.4</td>
<td>1434.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pr226</td>
<td>ACS</td>
<td>84534</td>
<td>86913.8</td>
<td>378.4</td>
<td>4065.25</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>83659</td>
<td>87215.6</td>
<td>523.8</td>
<td>5206.70</td>
</tr>
<tr>
<td></td>
<td>PACO</td>
<td>81967</td>
<td>83462.2</td>
<td>762.2</td>
<td>3103.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lin105</td>
<td>ACS</td>
<td>14883</td>
<td>15125.4</td>
<td>88.8</td>
<td>475.37</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>14795</td>
<td>15038.4</td>
<td>106.6</td>
<td>526.43</td>
</tr>
<tr>
<td></td>
<td>PACO</td>
<td>14736</td>
<td>14888</td>
<td>112.2</td>
<td>211.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kroB100</td>
<td>ACS</td>
<td>23014</td>
<td>23353.8</td>
<td>56.2</td>
<td>685.79</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>22691</td>
<td>23468.1</td>
<td>102.9</td>
<td>702.46</td>
</tr>
<tr>
<td></td>
<td>PACO</td>
<td>22289</td>
<td>22728</td>
<td>169.6</td>
<td>668.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kroC100</td>
<td>ACS</td>
<td>21594</td>
<td>21942.6</td>
<td>54.8</td>
<td>509.77</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>21236</td>
<td>21909.8</td>
<td>78.1</td>
<td>814.53</td>
</tr>
<tr>
<td></td>
<td>PACO</td>
<td>20775</td>
<td>21598.4</td>
<td>114.8</td>
<td>414.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lin318</td>
<td>ACS</td>
<td>48554</td>
<td>49224.4</td>
<td>849.2</td>
<td>1785.21</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>48282</td>
<td>49196.7</td>
<td>902.7</td>
<td>2459.16</td>
</tr>
<tr>
<td></td>
<td>PACO</td>
<td>47885</td>
<td>49172.8</td>
<td>866.8</td>
<td>1108.34</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the results obtained by GA-FDP [12], ACS-FDP [14] and the proposed ACO-FDP in FDP instances [14].

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>best</th>
<th>ave</th>
<th>time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem I</td>
<td>GA-FDP</td>
<td>4240.67</td>
<td>4261.4</td>
<td>153</td>
</tr>
<tr>
<td></td>
<td>ACS-FDP</td>
<td>4122.33</td>
<td>4138.5</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>ACO-FDP</td>
<td>4122.33</td>
<td>4126.2</td>
<td>80</td>
</tr>
<tr>
<td>Problem II</td>
<td>GA-FDP</td>
<td>4208</td>
<td>4250.6</td>
<td>184</td>
</tr>
<tr>
<td></td>
<td>ACS-FDP</td>
<td>4163</td>
<td>4289.2</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>ACO-FDP</td>
<td>4163</td>
<td>4165.8</td>
<td>135</td>
</tr>
</tbody>
</table>

The results in Table 2 indicate that PACO-FDP performs better than GA-FDP [12] and ACS-FDP [14] in the item of average length though it cannot find better solution than ACS-FDP [14]. PACO-FDP can be also considered as the improvement of ACS-FDP because the special strategies [14] are also used in PACO-FDP.

6. Discussions and Conclusions

This paper proposed an adaptive rule for volatility rate of pheromone trail, attempting to adjust the pheromone based on the solutions obtained by artificial ants. Thus, a new ACO algorithm is designed with this tuning rule. There is a special pheromone updating rule in the proposed algorithm whose framework is similar to Ant Colony System. Then, the convergence of the proposed ACO algorithm is proved to ensure its capacity of global capacity. Moreover, there are some experimental comparisons among the proposed ACO approach and other methods [2,12–14] in solving TSP and FDP problems. The results also show the effectiveness of the proposed algorithm.

Further study is suggested to explore the better management for the optimal setting of the parameters of ACO algorithms, which will be very helpful in the application.

7. Acknowledgements

This work has been supported by Natural Science Foundation of Guangdong Province (9151600301000001), Key Sci-tech Research Projects of Guangdong Province (2009B010800026), funded project of State Key Lab. for Novel Software Technology of Najing University and student research project (SRP) of South China University of Technology.
8. References


