Solving Quadratic Assignment Problems with the Cunning Ant System

Shigeyoshi Tsutsui, Member, IEEE and Lichi Liu

Abstract—In a previous paper, we proposed a variant of the ACO algorithm called the cunning Ant System (cAS) and evaluated it using TSP instances in the TSPLIB. The results showed that it could be one of the most promising ACO algorithms. In this paper, we applied cAS to solving the QAP and showed cAS has promising performance on the QAP as well. We introduced the entropy measure to evaluate the diversity of pheromone density and to analyze the convergence process of cAS. The results clearly showed evidence that the cunning scheme in cAS is effective in maintaining diversity of pheromone density and leads to successful search. The effectiveness of cAS was also confirmed when it was combined with the taboo local search.

I. INTRODUCTION

The quadratic assignment problem (QAP) is an NP-hard optimization problem and it is considered one of the hardest optimization problems [1], [2]. The QAP models many real-life problems arising in optimal location assignment problems, such as units assignment in a hospital, factory assignment of worldwide companies, etc. The QAP is also a good set of problems for testing the capabilities of solving combinatorial optimization problems.

There are many studies on solving QAP with ACO showing better results than with other meta-heuristics. These studies are summarized in [3]. Typical examples of ACO algorithms for the QAP are AS-QAP [4], MMAS-QAP [1] and ANTS-QAP [5]. Among these algorithms, ANTS-QAP and MMAS-QAP appear to perform significantly better than AS-QAP. MMAS-QAP is extended in [6].

In a previous paper [7], we have proposed a variant of the ACO algorithm called the cunning Ant System (cAS) and evaluated it using TSP instances in the TSPLIB. The results showed that it could be one of the most promising ACO algorithms. In this paper, we apply cAS to solving QAP and show cAS works well on the QAP instances in QAPLIB [8] as well. We also evaluate cAS when it is combined with Robust Taboo Search (Ro-TS) [9] using larger-size QAP instances in QAPLIB. The results also showed promising performance.

In the remainder of this paper, Section II gives a brief overview of cAS when it is applied in TSP. Then, Section III describes how the solutions with cAS for the QAP are constructed, and Section IV is empirical analysis of the cAS without heuristics and Section V is an empirical analysis of the cAS with heuristics. Finally, Section VI concludes this paper.

II. A BRIEF OVERVIEW OF cAS

cAS [7] introduced two important schemes. One is a scheme to use partial solutions which we call cunning. In constructing a new solution, cAS uses pre-existing partial solutions. With this scheme, we may prevent premature stagnation by reducing strong positive feedback to the trail density. The other is to use the colony model, dividing colonies into units, which has a stronger exploitation feature, while maintaining a certain degree of diversity among units. Using partial solutions to seed solution construction in the ACO framework has been performed by combining an external memory implementation in [10], [11]. In [6], some solution components generated according to ACO are removed, resulting in a partial candidate solution. Starting from the partial solution, a complete candidate solution is reconstructed by a greedy construction heuristic.

In cAS, we introduced the agent called cunning ant (c-ant). The c-ant differs from traditional ants in its manner of solution construction. It constructs a solution by borrowing a part of existing solutions. The remainder of the solution is constructed based on $\tau_{ij}(t)$ probabilistically as usual. In a sense, since this agent in part appropriates the work of others to construct a solution, we named the agent c-ant after the metaphor of its cunning behavior (in this paper a solution constructed by a c-ant is also represented with the same notation, c-ant)). Also, an agent from whom a partial solution has been borrowed by a c-ant is called a donor ant (d-ant) and the partial solution donated to the c-ant is also represented with the notation d-ant. Fig. 1 shows how c-ant works in the TSP.

Here note again the notations c-ant and d-ant are used both for agents and solutions. In this example, the c-ant borrows part of the tour, $\ldots \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3$, from the d-ant directly. The c-ant constructs the remainder of the tour for cities 4, 5, and 6 according to $\tau_{ij}(t)$ probabilistically. Using c-ant in this way, we may prevent premature stagnation the of search, because only a part of the cities in a tour are newly generated, and this may prevent over exploitation caused by strong positive feedback to $\tau_{ij}(t)$.

In cAS, we use a colony model as shown in Fig. 2, which is similar to the colony model proposed for real parameter optimization with ACO framework [12]. It consists of $m$ units. Each unit consists of only one $\text{ant}_{k,t}$ ($k = 1, 2, \ldots, m$). At iteration $t$, in unit $k$, a new $\text{ant}_{k,t+1}$ creates a solution with the existing ant in the unit (i.e., $\text{ant}_{k,t}$) as
of pheromone ant $C_k$ density $\tau$ [1]. Here, $\Delta \tau_{ij}$ is always reserved. Pheromone density $\tau_{ij}(t)$ is then updated with $\Delta \tau_{ij}^{k,t}$ is the amount of pheromone $C_{k,t}$ acts in QAP.

The colony model of $c$-ant $\phi$ in Eq. 5 related to both distances between locations and flows between facilities, the problem structure of QAP is much more complex and harder to solve than TSP [1], [2].

$\Delta* \tau_{ij}^{k,t} = 1/C_{k,t}^* : \text{if } (i,j) \in \text{ant}_{k,t}^* ; 0 : \text{otherwise}$, (2)

where the parameter $\rho$ ($0 \leq \rho < 1$) is the trail persistence (thus, $1-\rho$ models the evaporation), $\Delta* \tau_{ij}^{k,t}$ is the amount of pheromone $\text{ant}_{k,t}^*$ puts on the edge it has used in its tour, and $C_{k,t}$ is the fitness of $\text{ant}_{k,t}^*$.

$\tau_{ij}(t+1) = \rho \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta* \tau_{ij}^{k,t}$, (1)

$\text{unit 1}$

pheromone update

$\tau_{ij}(t+1)$

unit $m$

pheromone density

$\Delta* \tau_{ij}^{k,t} = 1/C_{k,t}^* : \text{if } (i,j) \in \text{ant}_{k,t}^* ; 0 : \text{otherwise}$, (2)

where the parameter $\rho$ ($0 \leq \rho < 1$) is the trail persistence (thus, $1-\rho$ models the evaporation), $\Delta* \tau_{ij}^{k,t}$ is the amount of pheromone $\text{ant}_{k,t}^*$ puts on the edge it has used in its tour, and $C_{k,t}$ is the fitness of $\text{ant}_{k,t}^*$.

$\tau_{ij}(t+1) = \rho \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta* \tau_{ij}^{k,t}$, (1)

$\text{unit 1}$

pheromone update

$\tau_{ij}(t+1)$

unit $m$

pheromone density

$\Delta* \tau_{ij}^{k,t} = 1/C_{k,t}^* : \text{if } (i,j) \in \text{ant}_{k,t}^* ; 0 : \text{otherwise}$, (2)

where the parameter $\rho$ ($0 \leq \rho < 1$) is the trail persistence (thus, $1-\rho$ models the evaporation), $\Delta* \tau_{ij}^{k,t}$ is the amount of pheromone $\text{ant}_{k,t}^*$ puts on the edge it has used in its tour, and $C_{k,t}$ is the fitness of $\text{ant}_{k,t}^*$.

$\tau_{ij}(t+1) = \rho \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta* \tau_{ij}^{k,t}$, (1)

$\text{unit 1}$

pheromone update

$\tau_{ij}(t+1)$

unit $m$

pheromone density

$\Delta* \tau_{ij}^{k,t} = 1/C_{k,t}^* : \text{if } (i,j) \in \text{ant}_{k,t}^* ; 0 : \text{otherwise}$, (2)

where the parameter $\rho$ ($0 \leq \rho < 1$) is the trail persistence (thus, $1-\rho$ models the evaporation), $\Delta* \tau_{ij}^{k,t}$ is the amount of pheromone $\text{ant}_{k,t}^*$ puts on the edge it has used in its tour, and $C_{k,t}$ is the fitness of $\text{ant}_{k,t}^*$.
C. Sampling methods

A crucial question when c-ant creates a new solution is how to determine which part of the solution the c-ant will borrow from the d-ant. To ensure robustness across a wide spectrum of problems, it should be advantageous to introduce variation both in the portion and the number of nodes of the partial solution that is borrowed from d-ant.

Let us represent the number of nodes that are constructed based on \( \tau_{ij}(t) \), by \( l_c \). Then, \( l_c \) is the number of nodes of partial solution, which c-ant borrows from d-ant, is \( l_c = n - l_s \). As in cAS in TSP, we use the control parameter \( \gamma \) which defines \( E(l_s) \) (the average of \( l_s \)) by \( E(l_s) = n \times \gamma \) and use the following probability density function \( f_s(l) \) used in [7] as

\[
f_s(l) = \begin{cases} \frac{1 - \gamma}{n - \gamma} (1 - \frac{l}{n})^{1 - \gamma} & \text{for } 0 < \gamma \leq 0.5, \\ \frac{\gamma}{n(1-\gamma)} (\frac{l}{n})^{\gamma - 1} & \text{for } 0.5 < \gamma < 1. \end{cases}
\]

Fig. 4 shows \( f_s(l) \) for \( \gamma = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, \) and 0.8.

In cAS for TSP, nodes in continuous positions of d-ant are copied to c-ant, because, in contrast to QAP, the partial solutions of d-ant are represented by nodes in continuous positions. In QAP it is not necessary for nodes to be continuous when c-ant borrows from d-ant or samples new nodes according to \( \tau_{ij}(t) \). Thus, in creating a new c-ant in QAP, nodes at some positions are copied and others are sampled with a random sequence of positions as follows: The number of nodes to be sampled \( l_s \) is generated by Eq. 7 with a given \( \gamma \) value. Then we copy the number of nodes, \( l_c = n-l_s \), from d-ant at random positions and sample the number of remaining nodes, \( l_s \), according to Eq. 6 with random sequence.

IV. PERFORMANCE OF CAS ON QAP

In this section, we evaluate cAS on the QAP using test instances in QAPLIB [8]. According to [1], [9], test instances can be classified into i) randomly generated instances, ii) grid-based distance matrix, iii) real-life instances, and iv) real-life-like instances. In this experiment, we use instances that are classified as real-life-like instances developed by Taillard, which have more relation with real world applications.

A. Performance of cAS on QAP

The performance of cAS was compared with MMAS [1]. The test instances are tai25b (n=25), tai30b (n=30), tai35b (n=35), and tai40b (n=40). The comparison was performed on the same number of solution constructions \( E_{max} = n \times 800,000 \). For number of units (or ants for MMAS) \( m = n \times 4 \) is used. \( \rho \) value of 0.9 and \( p_{best} \) value of 0.005 are used for both cAS and MMAS. 25 runs were performed.

Table 1 summarizes the results. The results of cAS are with \( \gamma \) value of 0.3. We also showed the results of non-cAS; i.e., we use the colony model shown in Fig. 2 but no cunning action is applied. This correspond to cAS with \( \gamma = 1 \). The code for MMAS is implemented by us and tuned to use the global best and the iteration best in pheromone update so as to get the smallest values of Error. We got the smallest value when we applied the global best every 5 iterations for pheromone update in MMAS. The pts strategy for MMAS was also tuned.

The values in bold-face show the best performance for each instance. From this table, we can see that cAS has good performance. Further, we can observe that even non-cAS has similar performance to MMAS and thus the effectiveness of using the colony model in Fig 2 is also confirmed as on TSP in [7].

Best_{avg} is average best solution over 25 runs and Error indicates average excess (%) of Best_{avg} from optimal in 25 runs.

B. The effect of \( \gamma \) values

Fig. 5 shows the variations of Error for various \( \gamma \) values. Here, \( \gamma \) values were varied starting from 0.1 to 0.9 with step
0.1. From this figure, we can see the effectiveness of using c-ant; i.e., with the smaller values of $\gamma$ (in the range of 0.1, 0.5)), the better values in Error are observed as cAS on TSP in [7]. To see the effect of $\gamma$ values on Error in more detail, here we do an analysis on the convergence process of the algorithm. As we discussed in Section I and II, the cunning action can be expected to prevent premature stagnation the of search, because only a part of the nodes in a solution are newly generated, and this could prevent over exploitation caused by strong positive feedback to $\tau_{ij}(t)$.

![Fig. 5. Variation of Error of cAS for $\gamma$.](image)

To see this effect, we introduce an entropy of pheromone density $\tau_{ij}(t)$ to measure the diversity of the system and analyze the convergence process of cAS in the following sub-section using the entropy measure.

**C. Analysis of the convergence process of cAS**

**1) Definition of entropy of pheromone density:** We define $I(t)$, entropy of pheromone density $\tau_{ij}(t)$, as follows:

$$I(t) = -\frac{1}{n} \sum_{t=0}^{n-1} \sum_{j=0}^{n-1} p_{ij}(t) \log p_{ij}(t),$$

where $p_{ij}(t)$ is defined as

$$p_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_{j=0}^{n} \tau_{ij}(t)}.$$  

This definition is slightly different from the entropy of $\tau_{ij}(t)$ defined for TSP [14] due to the difference of using pheromone density between TSP and QAP. But, it represents the diversity of $\tau_{ij}(t)$.

As is easily understood from the definition, the upper bound of $I(t)$ is obtained when all elements of $\tau_{ij}(t)$ have the same values as found during the initialization stage ($t=0$). This value is calculated as

$$T = \log(n).$$

To calculate the lower bound of $I(t)$, let’s consider an extreme case in which all strings have the same set of node values and have ants emit pheromones. If this iteration continues for a long time, all elements of $\tau_{ij}(t)$ converge to $\tau_{min}$ or $\tau_{max}$ as shown in Fig. 6.

![Fig. 6. Convergence of $\tau_{ij}(t)$ in QAP](image)
situations and can be calculated as Eq. 11 as follows:

\[ I \equiv \frac{\tau_{\text{max}}}{\tau_{\text{max}} + (n-1)\tau_{\text{min}}} \log \left( \frac{\tau_{\text{max}}}{\tau_{\text{max}} + (n-1)\tau_{\text{min}}} \right) - \frac{(n-1)\tau_{\text{min}}}{\tau_{\text{max}} + (n-1)\tau_{\text{min}}} \log \left( \frac{(n-1)\tau_{\text{min}}}{\tau_{\text{max}} + (n-1)\tau_{\text{min}}} \right) \]

(11)

where \( r = \frac{\tau_{\text{max}}}{\tau_{\text{min}}} \). In the following analysis, we use the normalized value of \( I_N(t) \) which is defined with \( I(t) \), \( T \), and \( I \) as

\[ I_N(t) = \frac{I(t) - I}{T - I} \]

(12)

Then, \( I_N(t) \) takes values in [0.0, 1.0].

2) The convergence process: The convergence processes for tai25b, tai30b, tai35b, and tai40b are presented in Figs. 7–10, respectively. In each figure, the left shows the change in Error and the right shows the change in Error. Each value in the figures shows averaged value over 25 independent runs.

On tai25 in Fig. 7, with \( \gamma \) values of 0.5, 0.7, and 0.9, we can see that \( I_N \) converges around at 80000, 40000, and 20000 iterations, respectively. These iterations coincide with the iterations where stagnations in Error occur. With \( \gamma \) value of 0.3, the value of \( I_N \) gradually decreases and the search continues with less stagnation. With \( \gamma \) value of 0.1, \( I_N \) keeps larger values until the end of run, resulting in slow convergence in Error. Figs. 8, 9, and 10 also show similar results for tai30b, tai35b, and tai40b, although their values in detail are different from each other.

From this convergence process analysis using the entropy measure, we can see the usefulness of the cunning scheme with smaller values of \( \gamma \). That is, on average, taking the rate of (1–\( \gamma \)) partial solution from existing solutions, and having the rate of \( \gamma \) partial solution being generated anew from the pheromone density can maintain diversity of the system, resulting in good balance between exploration and exploitation in the search. However, with extreme smaller values of \( \gamma \), i.e., \( \gamma \leq 0.1 \), the search processes becomes much slower, though the diversity of pheromone density can be maintained. Thus, we can see that choosing appropriate smaller \( \gamma \) values is important.

D. Alternative approach for determining the value of \( \ell_c \)

In the experiments in A.–C. of this section, the sampling nodes number \( \ell_c \) is determined probabilistically as described in Section III.C, using probability density function defined by Eq. 7. For example, let us consider the case of \( n=30 \) and \( \gamma=0.3 \). When we use Eq. 7 to determine the value of \( \ell_c \), it distributes in [0, 30] although \( E(\ell_c) \) \( (\text{the average value of } \ell_c) = n \times \gamma=9 \). For another option, we may determine the value of \( \ell_c \) as \( n \times \gamma=9 \) deterministically. In this subsection, we describe the results with this option.

Fig. 11 shows the change of Error for various \( \gamma \) for both of the above two cases. Here, deterministic represents the results with the new option and probabilistic represents the results with Eq. 7 (the results for this case are the same as the results shown in Fig. 5). From this figure, we can see...
that the values of Error in probabilistic are more robust to the change of $\gamma$ value than in deterministic, though choosing appropriate values of $\gamma$ with deterministic shows slightly better values of Error than probabilistic. Thus, we can say that using Eq. 7 for determining the value of $l_s$ in cAS is a good choice for robustness of the cAS algorithm.

V. PERFORMANCE OF cAS WITH LOCAL SEARCH

Here we study cAS with a local search on QAP. In [1], MMAS is combined with two local searches, i.e., Robust Taboo search algorithm (Ro-TS) developed by Taillard [9] and 2OPT. In this paper, we combined cAS with Ro-TS. Parameter settings and the methods of applying Ro-TS for cAS in this experiment are the same as were used in [1] as follows: $m$ value of 5, $p$ value of 0.8 and $p_{best}$ values of 0.005 are used. 250 times, short Ro-TS runs of length $4n$ are applied. (This setting was designed in [1] so that the computational time is the same as the Ro-TS carried out alone in which $1000 \times n$ iterations was allowed).

We used the Ro-TS code which is available at [15] though the code, which is originally written in C, was rewritten in JAVA since our cAS code is written in JAVA. Table II summarizes the results. All results except for cAS (MMAS-TS, MMAS-2OPT, GH-genetic hybrid, and Ro-TS) are taken from [1]. All the results in Table II are average values over 10 independent runs. Results of cAS is for $\gamma=0.8$. From these results, we can see that the cAS with Ro-TS works well. Variation of Error of cAS for $\gamma$ is shown in Fig. 12. Comparing Fig. 12 with Fig. 5, the better results are obtained at larger values of $\gamma$, i.e., $\gamma \geq 0.5$ in contrast to the cAS without heuristic, where with the smaller values of $\gamma$ cAS showed the better performance.

According to the results of MMAS with heuristic [1] MMAS-2OPT has better performance than MMAS-TS (please see Table 2). This fact is also described in [3]. Thus, we need to apply 2OPT to cAS, though this remains for future work.

TABLE II
RESULT OF cAS WITH RO-TS

<table>
<thead>
<tr>
<th>QAP</th>
<th>cAS ($\gamma=0, 8$)</th>
<th>non-cAS ($\gamma=1$)</th>
<th>MMAS-TS</th>
<th>MMAS-2OPT*</th>
<th>GH*</th>
<th>Ro-TS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>tai35b</td>
<td>0.00</td>
<td>0.00201</td>
<td>0.0051</td>
<td>0.0</td>
<td>0.107</td>
<td>0.064</td>
</tr>
<tr>
<td>tai40b</td>
<td>0.0</td>
<td>0.00001</td>
<td>0.0048</td>
<td>0.0</td>
<td>0.211</td>
<td>0.531</td>
</tr>
<tr>
<td>tai50b</td>
<td>0.00113</td>
<td>0.000241</td>
<td>0.172</td>
<td>0.009</td>
<td>0.214</td>
<td>0.342</td>
</tr>
<tr>
<td>tai60b</td>
<td>0.00091</td>
<td>0.000084</td>
<td>0.005</td>
<td>0.005</td>
<td>0.291</td>
<td>0.417</td>
</tr>
<tr>
<td>tai80b</td>
<td>0.00445</td>
<td>0.000574</td>
<td>0.591</td>
<td>0.266</td>
<td>0.829</td>
<td>1.031</td>
</tr>
<tr>
<td>tai100b</td>
<td>0.00155</td>
<td>0.00192</td>
<td>0.230</td>
<td>0.114</td>
<td>n.a.</td>
<td>0.512</td>
</tr>
</tbody>
</table>

* Data presented in [1]
VI. Conclusion

In this paper, we applied cAS to solving the QAP and showed cAS has a promising performance on QAP. We introduced the entropy measure to see the diversity of pheromone density and analyzed the convergence process of searches. The results clearly showed that the cunning scheme is effective in maintaining diversity of pheromone density and it leads to success in the search. Effectiveness of cAS is also confirmed when it was combined with the taboo search. But combining cAS with other local search, such as 2OPT remains for future work. We also used QAP test instances which are categorized as real-life-like instances. Study of cAS on other type of test instances also remains for future work.

Acknowledgment

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant-in-Aid for Scientific Research number 19500199.

References