Chinese Journal of Electronics
Vol.27, No.1, Jan. 2018

Performance Analysis of ACO on the Quadratic Assignment Problem*

XIA Xiaoyun[1] and ZHOU Yuren[2]

(1. *College of Mathematics Physics and Information Engineering, Jiaxing University, Jiaxing 314001, China*)

(2. *School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China*)

Abstract — The Quadratic assignment problem (QAP) is to assign a set of facilities to a set of locations with given distances between the locations and given flows between the facilities such that the sum of the products between flows and distances is minimized, which is a notoriously difficult NP-hard combinatorial optimization problem. A lot of heuristics have been proposed for the QAP problem, and some of them have proved to be efficient approximation algorithms for this problem. Ant colony optimization (ACO) is a general-purpose heuristic and usually considered as an approximation algorithms for NP-hard optimization problems. However, we know little about the performance of ACO for QAP from a theoretical perspective. This paper contributes to a theoretical understanding of ACO on the QAP problem. The worst-case bound on a simple ACO algorithm called (1+1) Max-min ant algorithm ((1+1) MMAA) for the QAP problem is presented. It is shown that a degenerate (1+1) MMAA finds an approximate solution on the QAP problem. Finally, we reveal that ACO can outperform the 2-exchange local search algorithm on an instance of the QAP problem.

Key words — Ant colony optimization (ACO), Quadratic assignment problem (QAP), Approximation algorithms, Local search, Algorithms analysis, Runtime analysis.

I. Introduction

Acolony optimization (ACO) is a popular stochastic heuristic optimization algorithm. This algorithm is inspired by the ability of some real ants species finding the shortest paths between their nest and a food source. The first ACO algorithm called Ant system (AS) was proposed by Dorigo and his colleagues[1] in 1991, which has been applied to solve the Traveling salesman problem (TSP). After that, ACO algorithm variants have been introduced and successfully applied to many combinatorial optimization and industrial problems[2].

In contrast to the many successful applications and empirical studies on benchmarks, the theoretical research on ACO algorithms is still limited. A major part of the theoretical study of ACO algorithms is the analysis on the runtime to find a global optimum for optimization problems[3].

Initially, the runtime analysis was carried out on the simple ACO algorithm for some artificial pseudo-Boolean functions such as OneMax, LeadingOnes, Trap function, and Linear function, *etc.* Gutjahr[4] and Neumann and Witt[5] independently proposed the runtime analysis of ACO algorithms on simple pseudo-Boolean functions. Neumann and Witt analyzed the running time of a simple ACO algorithm called the 1-ANT on OneMax, and showed that if the 1-ANT chooses an enough large evaporation factor ρ, then it behaves like the (1+1) EA[6]. Doerr *et al.*[7] further investigated the performance of the 1-ANT, and showed that if the evaporation factor is set too small, then the algorithm is easy to stagnate and the expected time until finding the global optimum is exponential for some simple pseudo-Boolean functions, *e.g.*, OneMax and LeadingOnes. A variant of ACO called Max-min ant system (MMAS)[8] with best-so-far update strategy can avoid stagnation and turn out to be efficient runtime on some functions. For example, with a scheme of ρ values decreasing, ACO shows polynomial runtime behavior on NH-OneMax problem[9]. It is also shown that MMAS with an appropriate ρ is efficient on OneMax and BinVal[10].

In the case of combinatorial optimization problems, some theoretical results of ACO on polynomial solvable

*Manuscript Received Mar. 15, 2016; Accepted Dec. 1, 2016. This work is supported by the National Natural Science Foundation of China (No.61703183, No.61773410), the Scientific Research Special Plan of Guangzhou Science and Technology Programme (No.201607010045), the Education Scientific Planning Project of Zhejiang Province (No.2017SCG048), and the Natural Science Foundation of Jiangxi Province of China (No.20151BAB217008).
 DOI:10.1049/cje.2017.06.004

problems have been obtained. Neumann and Witt[11] presented an analysis of ACO for the minimum spanning tree problem by using two different construction graphs. Zhou[12] analyzed the runtime behavior of ACO on two simple instances of the TSP, which is the first theoretical work of ACO for the NP-hard problem. Later, this work was significantly extended by Kötzing *et al.*[13] who also showed that ACO algorithms with two construction procedures can obtain a good approximation on random TSP instances in expected polynomial time. Attiratanasunthron and Fakcharoenphol[14] made a contribution to the theoretical understanding of the n-ANT algorithm for finding single destination shortest paths in Directed acyclic graphs (DAGs). Furthermore, Sudholt and Thyssen[15] significantly improved and generalized the results in Ref.[14], which can deal with arbitrary directed graphs containing cycles. Recently, the performance of ACO on dynamic fitness functions have been analyzed[16], which verified the effectiveness of ACO. Moreover, ACO algorithms have also been proven to be efficient for some noisy functions[17].

In practice, ACO is often used to obtain satisfying solutions. In this paper, we present an approximation performance analysis of the (1+1) MMAA on QAP problems. We demonstrate that the (1+1) MMAA can find an approximate solution on QAP problems with a cost not more than $\dfrac{\langle F, D\rangle_+}{s(F)s(D)} nC_{av}$ in expected runtime $O\left(n^7 \cdot \log \dfrac{s(F)s(D)}{2}\right)$, where C_{av} is the average cost among all possible solutions, $s(F)$ and $s(D)$ are the sums of all terms of two given positive integer matrices F and D, respectively, and $\langle F, D\rangle_+$ is the maximum scalar product of F and D. Finally, we construct a QAP instance, and show that the 2-exchange local search algorithm is easily trapped in a local optimum while the ACO algorithm can escape from it.

The rest of the paper is organized as follows. Section II introduces the QAP problem, and the algorithms considered in this paper. Section III analyzes the approximation performance of the (1+1) MMAA on QAP problems, and Section IV investigates the performance of the ACO algorithm for a constructed QAP instance. Finally, Section V presents conclusions.

II. Problem and Algorithm

1. The quadratic assignment problem

The QAP problem was formalized by Koopmans and Beckmann[18], which is a well-known combinatorial optimization problem and has been the subject of extensive research for over fifty years. From a practical point of view, this problem has been used in many fields such as backboard wiring, campus and hospital layout, typewriter keyboard design, scheduling problem, and many others can be formulated as QAPs. The QAP problem is proved to be an NP-hard optimization problem[19].

The QAP can be described as the problem of assigning a set of facilities to a set of locations with given distances between the locations and given flows between the facilities. Given a set of n facilities $F = \{F_1, F_2, \cdots, F_n\}$ and a set of n locations $D = \{D_1, D_2, \cdots, D_n\}$, a facility can be assigned to only one location, a location can hold only one facility, and all facilities must be assigned to a location. Let $F = (f_{ij})$ and $D = (d_{ij})$ be two $n \times n$ symmetric matrices, each having a null diagonal, where f_{ij} is the flow of materials from facility i to facility j, and d_{ij} represents the distance between location i and location j. We intend to find an assignment of all facilities to the locations that minimizes the total cost, *i.e.*, the sum of the products between flows and distances. Mathematically, the QAP problem aims to minimize the following objective function:

$$fit(\pi) = \sum_{i=1}^{n} \sum_{k=i+1}^{n} f_{ik} d_{\pi(i)\pi(k)} \tag{1}$$

where $\pi(i)$ corresponds to the location of facility i in permutation $\pi \in \mathit{\Pi}$, and $\mathit{\Pi}$ is the set of all permutations of $\{1, 2, \cdots, n\}$.

2. The ACO for QAP

For a given QAP instance, we construct a graph $G = (V, E)$, where V is the set of locations, *i.e.*, $V = \{1, 2, \cdots, n\}$, and E is the set of edges connecting each pair of locations. The artificial ant moves from vertex to vertex of V one by one, and finally generates a permutation $\pi \in \mathit{\Pi}$. The objective function Eq.(1) can guide ACO to find a permutation with the minimum total cost.

During the procedure of pheromone updating, the pheromones on the edges of the best-so-far solution are increased, while those on the other edges are decreased. This update procedure depends on the evaporation factor ρ which is a parameter of the ACO algorithm ($0 \le \rho \le 1$).

In this paper, we analyze the performance of a simple variant of ACO called the (1+1) Ant algorithm((1+1) AA) on the QAP problem, as shown in Algorithm 1.

Algorithm 1 (1+1) AA for QAP

```
1: Begin
2:    Initialization: Set parameters, pheromone values,
          choose an initial solution π;
3:    While termination criterion is not fulfilled
4:       Construct an ant solution π' by using
            construction procedure in Algorithm 2;
5:       Selection: If fit(π') < fit(π), then π = π';
6:       Update the pheromone values.
7:    End while
8: End
```

In the initialization step, each edge $(u, v) \in E$ randomly gets a pheromone value $\tau_{(u,v)}$, $u, v \in D$. Afterwards, a solution (a path in G) is constructed by the artificial ant starting from an arbitrary vertex $v \in V$ and visiting other vertices $V \setminus \{v\}$ one by one according to the construction procedure shown in Algorithm 2. The (1+1) AA uses a simple hill-climbing selection mechanism, and the current solution is replaced by a new one if this new solution is better than the current best-so-far solution. The pheromone values are updated according to some pheromone update mechanism.

Algorithm 2 Construction procedure

```
Begin
  Choose an arbitrary location v ∈ V as π(1), and
      mark v as visited;
  repeat
    Let N_v be the set of edges leading to non-visited
        successors of v in G;
    If N_v ≠ Ø then
      Choose e ∈ N_v with probability
        [τ_e]^α · [η_e]^β / Σ_{e'|e'∈N_v} [τ_e']^α · [η_e']^β;
      Let e = (v, w), mark w as visited, set v = w,
          and append e to π.
  until N_v = Ø
  return the constructed permutation π.
End
```

In the construction procedure, the two parameters α and β determine the relative importance of pheromone value and heuristic information. We use $\alpha = 1$ and $\beta = 0$ throughout this paper, which is a case investigated by Kötzing *et al.*[13].

To prevent premature convergence, we use the Max-min ant system[8] and restrict all pheromones to a bounded interval $[\tau_{\min}, \tau_{\max}]$, where $\tau_{\max}$ and $\tau_{\min}$ are the upper and lower bounds of pheromone values, respectively. Depending on whether edge $(\pi(i), \pi(j))$ is contained in the best-so-far solution π for $i, j = 1, 2, \cdots, n$, the pheromone update is performed as follows.

$$\tau'_{(\pi(i),\pi(j))} = \begin{cases} \min\{(1-\rho)\tau_{(\pi(i),\pi(j))} + \rho, \tau_{\max}\}, \\ \qquad\qquad \text{if } (\pi(i), \pi(j)) \in \pi \\ \max\{(1-\rho)\tau_{(\pi(i),\pi(j))}, \tau_{\min}\}, \\ \qquad\qquad \text{otherwise} \end{cases}$$

We call the (1+1) AA which uses this pheromone update mechanism the (1+1) MMAA.

The parameter ρ determines the strength of a pheromone update. The smaller the values of ρ, *i.e.*, lower evaporation, the smaller the pheromones change per iterations, and vice versa. For simplicity, we follow Kötzing et al.[13] to consider a degenerate case of the pheromone update mechanism in the approximation performance analysis of ACO on the QAP problem. That is the case of $\rho = 1$, *i.e.*, the pheromone update is simplified as follows.

$$\tau_{(\pi(i),\pi(j))} = \begin{cases} \tau_{\max}, & \text{if } (\pi(i), \pi(j)) \in \pi \\ \tau_{\min}, & \text{otherwise} \end{cases}$$

Further, we choose $\tau_{\min} = \frac{1}{n^2}$ and $\tau_{\max} = 1 - \frac{1}{n}$ as Zhou[12] and Kötzing et al.[13] did. This choice is inspired by standard mutations in evolutionary computations where each bit has a probability of $\frac{1}{n}$ to flip, which can ensure to rediscover the previous path with constant probability.

In this paper, the complexity results of runtime analysis refer to the number of steps for constructing solutions until that an optimal or an approximate solution is obtained.

III. Approximation Performance Analysis of the (1+1) MMAA on the QAP Problem

In order to investigate the performance guarantee of the (1+1) MMAA for the QAP problem, we first introduce a local search operator called 2-exchange under the set Π of permutations of $\{1, 2, \cdots, n\}$. It can be described as follows.

A permutation $\pi = (\pi(1), \cdots, \pi(i-1), \pi(i), \pi(i+1), \cdots, \pi(j-1), \pi(j), \pi(j+1), \cdots, \pi(n))$ is transformed into $\pi' = (\pi(1), \cdots, \pi(i-1), \pi(j), \pi(i+1), \cdots, \pi(j-1), \pi(i), \pi(j+1), \cdots, \pi(n))$ by exchanging two locations i and j.

We now briefly recall the basic principles of the local search method, and give some notations and related techniques, which have been used by Angel and Zissimopoulos[20]. Let $s(F)$ be the sum of all terms of a given facility matrix F, *i.e.*, $s(F) = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}$. Similarly, $s(D) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}$. Let x and y be two vectors of the same dimension. The maximum scalar product of x and y is defined by: $\langle x, y\rangle_+ = \max_{\pi \in \Pi}\langle x, \pi y\rangle$. Let F_k and D_k denote the sum over the kth column of F and D, respectively, and $\langle F, D\rangle_+$ be an abbreviation for $\langle (F_1, \cdots, F_n), (D_1, \cdots, D_n)\rangle_+$. For a given QAP instance, if there exists an exchange between location i and location j such that the total cost can be decreased, we call it a good step. The average cost C_{av} and maximum cost $C_{\max}$ among all possible permutations are proven to be $\frac{s(F)s(D)}{2n(n-1)}$ and less than $\frac{s(F)s(D)}{2}$ by Angel and Zissimopoulos[20], respectively, *i.e.*, $C_{av} = \frac{s(F)s(D)}{2n(n-1)}$ and $C_{\max} \le \frac{s(F)s(D)}{2}$.

In the case of $\rho = 1$, we reveal that the ACO algorithm starting from any arbitrary solution can efficiently achieve an approximate solution on QAP problems with a cost not more than $\frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}$, which is the worst performance guarantee of a local search algorithm proven by Angel and Zissimopoulos[20]. The main idea is that the ACO algorithm can simulate the local search algorithm with 2-exchange neighborhood.

According to the proof of Theorem 1 in Ref.[20], if a solution π is not a local optimum, then there exists a 2-exchange operation which improves the cost by producing a new solution π^*. Furthermore, the cost of π^* satisfies the following inequality.

$$fit(\pi^*) \leq fit(\pi) - \frac{4}{n}(fit(\pi) - C_{av}) + \varphi \qquad (2)$$

where $\varphi = \max_{\pi} \frac{2}{n(n-1)}(s(F \cdot \pi(D)) - \frac{s(F)s(D)}{n})$, with the matrix $\pi(D)$ defined by $(\pi(D))_{ij} = d_{\pi(i)\pi(j)}$. Iterating this process gives a limit cost l that satisfies $l = l - \frac{4}{n}(l - C_{av}) + \varphi$. It follows that $l = \frac{\varphi n}{4} + C_{av}$.

We denote the 2-exchange neighborhood of π by $N(\pi)$, which includes all permutations produced by 2-exchange operation on π. It is clear that $N(\pi)$ contains $\frac{n(n-1)}{2}$ permutations. Note that the total number of permutations is $n!$. Since the search space is extremely large, it may need exponential time to find the optimal permutation by evaluating them all. Therefore, we consider the approximate permutation which can be obtained within a reasonable time.

We partition the set Π into two disjoint subsets: the approximation subset $S_1 = \{\pi \in \Pi \mid fit(\pi) \leq \frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}\}$ and its complement $S_2 = \{\pi \in \Pi \mid fit(\pi) > \frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}\}$. And we denote the sequence of permutation by $\pi_t (t = 0, 1, \cdots)$, where π_t is the permutation produced by the (1+1) MMAA at iteration t.

Lemma 1 Let $\alpha = 1$ and $\beta = 0$, $\tau_{\min} = \frac{1}{n^2}$ and $\tau_{\max} = 1 - \frac{1}{n}$. If $\pi_t \in S_2$, for $\rho = 1$, then with probability $\Omega(\frac{1}{n^6})$, the (1+1) MMAA reaches a solution $\pi' \in N(\pi_t)$ at iteration $t+1$ such that $fit(\pi') \leq fit(\pi_t) - \frac{4}{n}(fit(\pi_t) - C_{av}) + \varphi$.

Proof Let $\pi_t = \pi = (\pi(1), \cdots, \pi(i), \cdots, \pi(j), \cdots, \pi(n))$. Since $\pi_t \in S_2$, by Eq.(2), there exists $\pi' = (\pi(1), \cdots, \pi(j), \cdots, \pi(i), \cdots, \pi(n)) \in N(\pi)$ such that $fit(\pi') \leq fit(\pi_t) - \frac{4}{n}(fit(\pi_t) - C_{av}) + \varphi$. Fig.1 describes the 2-exchange operation which changes from π to π'.

For simplicity, we assume $n > 8$. The key of the proof lies in the estimate of the transition probability that the (1+1) MMAA transforms π to π'. We denote A as the event that the artificial ant has finished the path π_t at iteration t, and denote E_k $(k = 1, 2, \cdots, 8)$ as the events that the artificial ant walks on the path $(\pi(1), \pi(2), \cdots, \pi(i-1))$ from $\pi(1)$ to $\pi(i-1)$, the path $(\pi(i-1), \pi(j))$ from $\pi(i-1)$ to $\pi(j)$, the path $(\pi(j), \pi(i+1))$ from $\pi(j)$ to $\pi(i+1)$, the path $(\pi(i+1), \pi(i+2), \cdots, \pi(j-1))$ from $\pi(i+1)$ to $\pi(j-1)$, the path $(\pi(j-1), \pi(i))$ from $\pi(j-1)$ to $\pi(i)$, the path $(\pi(i), \pi(j+1))$ from $\pi(i)$ to $\pi(j+1)$, the path $(\pi(j+1), \pi(j+2), \cdots, \pi(n-1))$ from $\pi(j+1)$ to $\pi(n-1)$, and the path $(\pi(n-1), \pi(n))$ from $\pi(n-1)$ to $\pi(n)$ respectively.

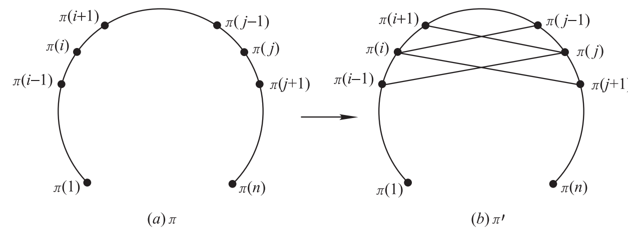

Fig. 1. A 2-exchange operation

In the following, we estimate the conditional probability of event E_i given $\bigcap_{j=1}^{i-1} E_j (1 \leq j \leq i \leq 8)$ and A, *i.e.*, $P(E_i \mid \bigcap_{j=1}^{i-1} E_j A (1 \leq j \leq i \leq 8))$. The ant processes through eight stages, for any $l \in \{1, 2, \cdots, n-1\}$, we have:

Stage 1: When the ant is on a vertex l which is on the sub-path from $\pi(1)$ to $\pi(i-2)$, there are at most $n-1$ edges incident to l. Since the vertices visited can't be chosen, there is one with pheromone value $1 - \frac{1}{n}$ and at most $n-2$ with pheromone value $\frac{1}{n^2}$. According to Algorithm 2, we have

$$P(E_1 \mid A) \geq \left(\frac{1 - \frac{1}{n}}{(1 - \frac{1}{n}) + (n-2)\frac{1}{n^2}} \right)^{i-2}$$

Stage 2: When the ant is on vertex $\pi(i-1)$, there are $n-1$ edges incident to $\pi(i-1)$. Since the vertices visited can't be chosen, there is one with pheromone value $1 - \frac{1}{n}$ and at most $n-2$ with pheromone value $\frac{1}{n^2}$. Hence

$$P(E_2 \mid E_1 A) \geq \frac{\frac{1}{n^2}}{(1 - \frac{1}{n}) + (n-2)\frac{1}{n^2}} \geq \frac{1}{n^2}$$

Stage 3: When the ant is on vertex $\pi(j)$. Since the edge $(\pi(i-1), \pi(j))$ has been visited, there are two with pheromone value $1 - \frac{1}{n}$ and at most $n-4$ edges with pheromone value $\frac{1}{n^2}$ left to be visited. So

$$P(E_3 \mid E_1E_2A) \geq \frac{\frac{1}{n^2}}{2(1-\frac{1}{n})+(n-4)\frac{1}{n^2}} \geq \frac{1}{2n^2}$$

Stage 4: When the ant is on a vertex l which is on the sub-path from $\pi(i+1)$ to $\pi(j-2)$, this stage is similar to stage 1. Hence

$$P(E_4 \mid E_1E_2E_3A) \geq \left(\frac{1-\frac{1}{n}}{(1-\frac{1}{n})+(n-2)\frac{1}{n^2}}\right)^{j-i-2}$$

Stage 5: When the ant is on vertex $\pi(j-1)$. Since the vertices $\pi(j-2)$ and $\pi(j)$ have been visited, there are at most $n-3$ edges with pheromone value $\frac{1}{n^2}$ left to be visited, we have

$$P(E_5 \mid E_1E_2E_3E_4A) \geq \frac{\frac{1}{n^2}}{(n-3)\frac{1}{n^2}} \geq \frac{1}{n}$$

Stage 6: When the ant is on vertex $\pi(i)$. Since the vertices $\pi(i-1)$, $\pi(i+1)$ and $\pi(j-1)$ have been visited, there are at most $n-4$ edges with pheromone value $\frac{1}{n^2}$ left to be visited. Therefore

$$P(E_6 \mid E_1E_2E_3E_4E_5A) \geq \frac{\frac{1}{n^2}}{(n-4)\frac{1}{n^2}} \geq \frac{1}{n}$$

Stage 7: When the ant is on a vertex l which is on the sub-path from $\pi(j+1)$ to $\pi(n-2)$, this stage is similar to stage 1, so

$$\begin{aligned}&P(E_7 \mid E_1E_2E_3E_4E_5E_6A)\\ &\geq \left(\frac{1-\frac{1}{n}}{(1-\frac{1}{n})+(n-2)\frac{1}{n^2}}\right)^{n-j-2}\end{aligned}$$

Stage 8: The ant is on vertex $\pi(n-1)$, only the vertex $\pi(n)$ has not been visited, the ant will visit the last edge $(\pi(n-1),\pi(n))$ and complete the path, we have

$$P(E_8 \mid E_1E_2E_3E_4E_5E_6E_7A) = 1$$

Combining Stages 1–8, we conclude that the probability for the (1+1) MMAA to perform such a 2-exchange operation from π to π' is at least

$$\begin{aligned}&P(\pi_{t+1}=\pi' \mid \pi_t=\pi)\\
&=P(\bigcap_{i=1}^{8} E_i \mid A)\\
&=P(E_1 \mid A)P(E_2 \mid E_1A)P(E_3 \mid E_1E_2A)\\
&\quad\cdot P(E_4 \mid E_1E_2E_3A)P(E_5 \mid E_1E_2E_3E_4A)\\
&\quad\cdot P(E_6 \mid E_1E_2E_3E_4E_5A)P(E_7 \mid E_1E_2E_3E_4E_5E_6A)\\
&\quad\cdot P(E_8 \mid E_1E_2E_3E_4E_5E_6E_7A)\\
&\geq \left(\frac{1-\frac{1}{n}}{(1-\frac{1}{n})+(n-2)\frac{1}{n^2}}\right)^{i-2}\cdot\frac{1}{n^2}\cdot\\
&\quad\cdot\frac{1}{2n^2}\cdot\left(\frac{1-\frac{1}{n}}{(1-\frac{1}{n})+(n-2)\frac{1}{n^2}}\right)^{j-i-2}\\
&\quad\cdot\frac{1}{n}\cdot\frac{1}{n}\cdot\left(\frac{1-\frac{1}{n}}{(1-\frac{1}{n})+(n-2)\frac{1}{n^2}}\right)^{n-j-2}\cdot 1\\
&\geq \frac{1}{2\cdot n^6}\cdot\left(\frac{1-\frac{1}{n}}{(1-\frac{1}{n})+(n-2)\frac{1}{n^2}}\right)^{n-6}\\
&\geq \frac{1}{2\cdot n^6}\cdot(1-\frac{1}{n})^{n-6}\\
&\geq \frac{1}{2\cdot e\cdot n^6}. \quad (\text{Since } (1-\frac{1}{n})^{n-6}\geq\frac{1}{e},\ e=2.7182\ldots)\end{aligned}$$

Altogether we complete the proof.

Let $X_t = fit(\pi_t)$. We denote by Y_t the difference between the cost of the current assignment and the limit cost l at iteration t, *i.e.*, $Y_t = X_t - l = fit(\pi_t) - l$ and denote by $E[Y_t]$ the expectation of Y_t at iteration t. We hope that Y_t tends to zero step by step via the execution of 2-exchange operations in sequence of permutations. And we need to estimate the expected one-step difference decrease. The detailed statement can be described as follows.

Lemma 2 If $Y_t \geq 0$, then $E[Y_{t+1} \mid Y_t] \leq (1-\frac{c}{n^7})Y_t$, where $Y_t = fit(\pi_t) - l$, and c is a constant.

Proof Let p be the conditional probability that the (1+1) MMAA improves the solutions at iteration $t+1$ given Y_t. According to the definition of expectation, the expected difference between the cost of the current assignment and l consists of two parts: accepted case (with good step), and non-accepted case.

Since $l = \frac{\varphi n}{4} + C_{av}$, we have

$$\begin{aligned}&E[Y_{t+1} \mid Y_t]\\
&=E[X_{t+1}-l \mid X_t]\\
&\leq p(X_t-\frac{4}{n}(X_t-C_{av})+\varphi-l)+(1-p)(X_t-l)\\
&\qquad\qquad(\text{By expectation definition and Lemma 1})\\
&=p(X_t-\frac{4}{n}(X_t-C_{av})+\varphi-(\frac{\varphi n}{4}+C_{av}))+(1-p)(X_t-l)\\
&=p(1-\frac{4}{n})(X_t-\frac{\varphi n}{4}-C_{av})+(1-p)(X_t-l)\\
&=(1-\frac{4p}{n})(X_t-l)\\
&\leq(1-\frac{c}{n^7})Y_t. \quad (\text{By Lemma 1, } p=\Omega(\frac{1}{n^6}),\ c \text{ is a constant})\end{aligned}$$

For the local search algorithm with 2-exchange neighborhood, Angel and Zissimopoulos[20] proved that this algorithm can find an approximate solution with a cost not more than $\frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}$ for the QAP problem.

Lemma 3 (Angel and Zissimopoulos[20]) For the QAP problem, the limit cost l obtained by the local search algorithm around the 2-exchange satisfies $l \leq \frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av} \leq \frac{n}{2}C_{av}$.

Based on the above discussion, the (1+1) MMAA can improve the value of the objective function by performing a 2-exchange operation with probability $\Omega(\frac{1}{n^6})$. So, by Lemma 3, we can analyze the expected time for the (1+1) MMAA to find a local optimum with a cost not more than $\frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}$ for QAP problems by simulating the local search around the 2-exchange neighborhood. In order to analyze the runtime, we would like to evaluate the best-so-far solutions during the execution $\pi_0, \pi_1, \pi_2, \cdots$ as an infinite stochastic process.

Theorem 1 Let $\alpha = 1$ and $\beta = 0$, $\tau_{\min} = \frac{1}{n^2}$ and $\tau_{\max} = 1 - \frac{1}{n}$. For any QAP instance, if the entries of the matrices F and D are positive integers, then the (1+1) MMAA with $\rho = 1$ can reach a solution with a cost not more than $\frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}$ in expected running time $O(n^7 \cdot \log \frac{s(F)s(D)}{2})$.

Proof According to Lemma 2, we can estimate the expected time of the (1+1) MMAA starting from any permutation to find a local optimum with a cost not more than $\frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}$.

$$\begin{aligned}
&E[Y_t \mid \pi_0, \pi_1, \cdots, \pi_{t-1}] \\
&\quad = E[X_t - l \mid \pi_0, \pi_1, \cdots, \pi_{t-1}] \\
&\quad \leq (1-\frac{c}{n^7})E[X_{t-1} - l \mid \pi_0, \pi_1, \cdots, \pi_{t-2}] \text{ (by Lemma 2)} \\
&\quad \leq (1-\frac{c}{n^7})^2 E[X_{t-2} - l \mid \pi_0, \pi_1, \cdots, \pi_{t-3}] \\
&\quad \cdots\cdots \\
&\quad \leq (1-\frac{c}{n^7})^t E[X_0 - l] \quad \text{(By induction)}
\end{aligned}$$

Since $X_0 - l \leq C_{\max} \leq \frac{s(F)s(D)}{2}$, we have

$$\begin{aligned}
&E[Y_t \mid \pi_0, \pi_1, \cdots, \pi_{t-1}] \\
&\quad \leq (1-\frac{c}{n^7})^t E[X_0 - l] \\
&\quad \leq (1-\frac{c}{n^7})^t \cdot C_{\max}
\end{aligned}$$

Set $T = \frac{\log 2 + \log C_{\max}}{c} \cdot n^7$. Then for $t \geq T$, we have

$$E[Y_t \mid \pi_0, \pi_1, \cdots, \pi_{t-1}] \leq \frac{1}{2}$$

According to Markov's inequality, after T steps, the probability that Y_t is at least one is

$$P(Y_t \geq 1) \leq \frac{E[Y_t]}{1} \leq \frac{1}{2}$$

So, $P(Y_t < 1) \geq \frac{1}{2}$, *i.e.*, after T steps, the probability that the distance of Y_t is less than one is at least $\frac{1}{2}$. By Lemma 3, since all possible values of the objective function contain only positive integers, this implies that after T steps, the probability that the (1+1) MMAA finds a solution with a cost not more than $\frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}$ is at least $\frac{1}{2}$. Therefore, the expected runtime that the (1+1) MMAA reaches a solution with a cost not more than $\frac{\langle F,D\rangle_+}{s(F)s(D)}nC_{av}$ is bounded from above by $2T = (2\frac{\log 2 + \log C_{\max}}{c} \cdot n^7) = O(n^7 \cdot \log C_{\max}) = O(n^7 \cdot \log \frac{s(F)s(D)}{2})$. This completes the proof.

As shown in Theorem 1, the (1+1) MMMA can find the same approximation solution as the 2-exchange local search algorithm. We have made a step towards a better understanding of the approximation performance of ACO algorithms on an NP hard problem.

IV. The Analysis of the (1+1) MMAA on a QAP Instance

Local search algorithm as an iterative improvement method, it starts with an initial solution and then searches the solution space by moving from a current solution to a better solution in its neighborhood.

In this section, we present a QAP instance Q_1 and demonstrate that iterative improvement algorithms may terminate in a local optimum, but the ACO can solve this instance. We analyze the runtime for a more general pheromone model which more closely reflects the ACO paradigm with $\rho \neq 1$ on a constructed QAP instance. The rigorous theoretical analysis shows that there are cases where ACO performs better than the local search algorithm with 2-exchange neighborhood. Note that the 2-opt replaces two edges, while the 2-exchange operation swaps two vertices (locations), which can be seen as replacing four edges, *i.e.*, two 2-opts.

We set an $n \times n$ symmetric matrix $F = (f_{ij})$ of Q_1 as

$$f(i,j) = \begin{cases} 0, & \text{if } i = j \\ n^4, & \text{if } |i-j| = 1 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

i.e.,

$$F = \begin{pmatrix} 0 & n^4 & 1 & \cdots & 1 & 1 & 1 \\ n^4 & 0 & n^4 & \cdots & 1 & 1 & 1 \\ 1 & n^4 & 0 & \cdots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 0 & n^4 & 1 \\ 1 & 1 & 1 & \cdots & n^4 & 0 & n^4 \\ 1 & 1 & 1 & \cdots & 1 & n^4 & 0 \end{pmatrix}$$

and set an $n \times n$ symmetric matrix $D = (d_{ij})$ of Q_1 as

$$d(i,j) = \begin{cases} 0, & \text{if } i = j \\ \dfrac{1}{n^4}, & \text{if } |i-j| = 1 \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

i.e.,

$$D = \begin{pmatrix} 0 & \frac{1}{n^4} & 1 & \cdots & 1 & 1 & 1 \\ \frac{1}{n^4} & 0 & \frac{1}{n^4} & \cdots & 1 & 1 & 1 \\ 1 & \frac{1}{n^4} & 0 & \cdots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 0 & \frac{1}{n^4} & 1 \\ 1 & 1 & 1 & \cdots & \frac{1}{n^4} & 0 & \frac{1}{n^4} \\ 1 & 1 & 1 & \cdots & 1 & \frac{1}{n^4} & 0 \end{pmatrix}$$

According to matrix D, the distance of each edge is $\frac{1}{n^4}$ (light edges) or 1 (heavy edges). For simplicity, we assume n is even. Given a path π of locations, if it has k edges with distance 1, according to the definition of the objective function in Eq.(1), the sum of cost for this instance includes $\frac{n(n-1)}{2}$ items. These items contains four kinds of products of flows and distances, *i.e.*, $n^4 \cdot 1$, $n^4 \cdot \frac{1}{n^4}$, $1 \cdot \frac{1}{n^4}$, $1 \cdot 1$. The number of items which contain n^4 and $\frac{1}{n^4}$ are $n-1$, since there are k edges of distance 1, and one edge with distance 1 represents an item of 1. Therefore, there are k products of n^4 and 1, $n-1-k$ products of n^4 and $\frac{1}{n^4}$, k products of 1 and $\frac{1}{n^4}$, and $\frac{n(n-1)}{2} - (n-1+k)$ products of 1 and 1. Then the cost $fit(\pi)$ of π is $k(n^4 \cdot 1) + (n-1-k)(n^4 \cdot \frac{1}{n^4}) + k(1 \cdot \frac{1}{n^4}) + (\frac{n(n-1)}{2} - (n-1+k))(1 \cdot 1)$. By simplifying, we have $fit(\pi) = k(n^2 - \frac{1}{n^2})^2 + \frac{n(n-1)}{2}$ $(k = 0, 1, \cdots, n-1)$. There are altogether n different distances of the paths in Q_1. It has exactly one optimal permutation$(1, 2, \cdots, n-1, n)$, where the edges between the location i and $i+1$ are light edges, and the others are heavy edges, see Fig.2(a). The total cost of this optimal permutation is $\frac{n(n-1)}{2}$.

The local optimal permutation $\pi = (2, 1, 4, 3, \cdots, n, n-1)$ for the 2-exchange local search algorithm on Q_1 is depicted in Fig.2(b), which is obtained from the optimal permutation in Fig.2(a) by exchanging the locations $2i$ and $2i-1$ for $i = 1, 2, \cdots, \frac{n}{2}$. We can see that the 2-exchange operation, which exchanges any two locations in this permutation, can not decrease the value of the objective function. Thus, the 2-exchange local search algorithm may get trapped in this local optimum.However, the (1+1) MMAA can escape from this local optimum.

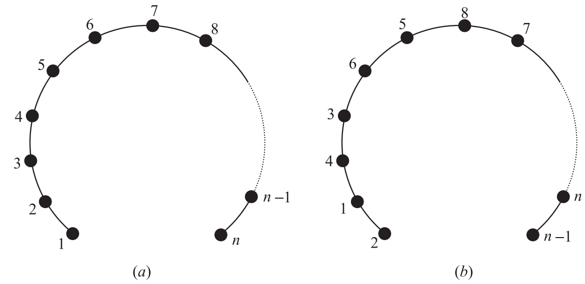

Fig. 2. (a) The global optimal permutation for Q_1; (b) The local optimal permutation for the 2-exchange local search algorithm on Q_1

We introduce the level-reaching estimations technique [21], which has been developed and successfully used in the analysis of evolutionary algorithms.

Definition 1 (Fitness-based partitions) Given a finite search space S, we consider a function $f : S \rightarrow R$ to be minimized. For two disjoint sets $A, B \subseteq S$, we say $A <_f B$ if $f(a) < f(b)$ for all $a \in A$ and all $b \in B$. S is partitioned into disjoint sets $A_0, A_1, \cdots, A_m$ such that $A_0 <_f A_1 <_f A_2 <_f \cdots <_f A_m$. We call $A_i = \{x \in S \mid f(x) = f_i\}$ the f-based partition for $i = 0, 1, \cdots, m$.

For Q_1, let $A_k = \{\pi \mid \pi$ is a permutation of locations, $fit(\pi) = k(n^2 - \frac{1}{n^2})^2 + \frac{n(n-1)}{2}\}$ for $k = 0, 1, \cdots, n-1$. Then A_0 contains the only optimal permutation, and $A_0, A_1, \cdots, A_{n-1}$ are fitness-based partitions of the search space.

Lemma 4[9] Given an f-based partition, let $\pi_t (t = 1, 2, \cdots)$ be the path produced by the (1+1) MMAA for QAP at iteration t, and $p_i (i = 1, 2, \cdots, m)$ be a lower bound on the probability of π_t leaving set A_i to $A_{i-1} \cup \cdots \cup A_0$. A_0 contains only optimal search points. Then the expected optimization runtime of the (1+1) MMAA on f to find the optimal search points is bounded from above by $\sum_{i=1}^{m} (s_i^* + \frac{1}{p_i})$, where s_i^* is the freezing time.

The freezing time s_i^* is a random time until all pheromones have reached their borders, *i.e.*, the time to bring all pheromone values to upper or lower bounds, which was first used by Neumann and Witt[10]. Following Gutjahr and Sebastiani [9], the freezing time satisfies $s_i^* \leq \frac{-\log(n-1)}{\log(1-\rho)}$. By using $\log(1-\rho) \leq -\rho$, for $0 \leq \rho \leq 1$, we have $s_i^* \leq (\log n)/\rho$. So, the expected runtime for the (1+1) MMAA is bounded from above by

$$\sum_{i=1}^{m} (\frac{\log n}{\rho} + \frac{1}{p_i}) \tag{5}$$

Lemma 4 provides an expected runtime estimation of the simple ACO algorithm which was proposed by Gutjahr and Sebastiani[9].

Zhou[12] first investigated the runtime of ACO algorithms for the TSP problem. He analyzed the runtime of the (1+1) MMAA on two simple instances. Based on that work by Zhou[12], Kötzing *et al.*[13] further investigated a new edge-based construction graph which has a stronger local property than the commonly used one for constructing solutions of TSP problems. They proved that the expected runtime $O(n^6 + \frac{n \log n}{\rho})$ of the (1+1) MMAA for G_1 in Zhou[12] can be improved to $O(n^3 \log n + \frac{n \log n}{\rho})$.

The following theorem presents a lower bound of $\Omega(\frac{1}{n^5})$ on p_i, the probability of the (1+1) MMAA leaving set A_k for instance Q_1.

Theorem 2 Let $\alpha = 1$ and $\beta = 0$, $\tau_{\min} = \frac{1}{n^2}$ and $\tau_{\max} = 1 - \frac{1}{n}$, and $\pi_t (t = 1, 2, \cdots)$ the best-so-far path sequence found by the (1+1) MMAA on Q_1 at iteration $t > 0$. If π_t is not the optimal, then $\pi_t \in A_i (i > 0)$, and the probability of an improvement satisfies $p_i = P(\pi_{t+1} \in A_{i-1} \cup \cdots \cup A_0 \mid \pi_t \in A_i) = \Omega(\frac{1}{n^5})$.

Proof For simplicity, we assume $n > 6$. Suppose the (1+1) MMAA constructs an initial permutation $\pi_t = \pi = (\pi(1), \cdots, \pi(i), \cdots, \pi(j), \cdots, \pi(n)) \in A_i$ of n locations on Q_1 randomly. If the current path π_t is not the optimal, then there must exist a light edge $e = (\pi(i), \pi(j)) \notin \pi_t$. Each location is incident to two light edges, and both $\pi(i)$ and $\pi(j)$ are incident to exactly one light edge different from edge e. Because edge $e \notin \pi_t$, there must exist two heavy edges $e_0 \in \pi_t$ and $e_1 \in \pi_t$ which are incident to $\pi(i)$ and $\pi(j)$, respectively. We denote the other two edges $e_0' \in \pi_t$ and $e_1' \in \pi_t$ which are incident to $\pi(i)$ and $\pi(j)$, respectively. The (1+1) MMAA intends to obtain an improvement of π which contains the edges e, e_0', e_1' but not the heavy edges e_0 and e_1. We distinguish two cases: case 1 is that e_0 and e_1 are on the two opposite sides of the light edge e, and case 2 is that e_0 and e_1 are on the same side of the light edge e, see Fig.3.

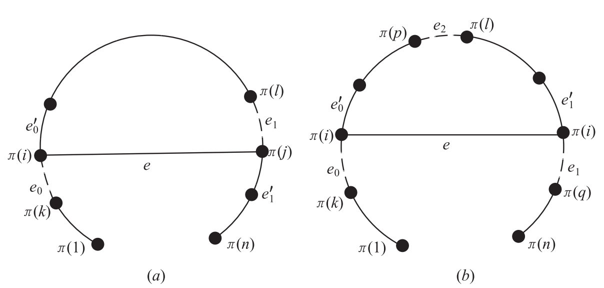

Fig. 3. (a) e_0 and e_1 are on the two opposite sides of the light edge e; (b) e_0 and e_1 are on the same side of the light edge e

Case 1: In this case, we replace two edges $\{(\pi(k), \pi(i)) = e_0, (\pi(l), \pi(j)) = e_1\}$ of π with two other edges $\{(\pi(k), \pi(l)), (\pi(i), \pi(j)) = e\}$, which produce a modified permutation $\pi' = (\pi(1), \cdots, \pi(k), \pi(l), \cdots, \pi(i), \pi(j), \cdots, \pi(n))$, see Fig.3(a). Since $d(\pi(k), \pi(i)) = d(\pi(l), \pi(j)) = 1$ and $d(\pi(i), \pi(j)) = \frac{1}{n^4}$, the cost of π' is better than that of the current path π. We consider the transition probability that the ant switches from π to π'. Following Zhou[12], we need to implement 2-opt move which can obtain an improvement, and the probability of such an improvement is $\Omega(\frac{1}{n^3})$.

Case 2: We claim that there must exist at least one heavy edge in the path $(\pi(i), e_0', \cdots, e_1', \pi(j))$ of π. Suppose otherwise, there exists a cycle $(\pi(i), e_0', \cdots, e_1', \pi(j), \pi(i))$ containing only light edges. Since all vertices on this cycle are incident with two light edges, they can not be incident to any vertex which is not on this cycle by a light edge but a heavy edge. Thus, we could not construct a path consisting of all light edges, which contradicts the fact that there exists a path consisting of all light edges. Let $(\pi(p), \pi(l)) = e_2$ be the first heavy edge. We replace three edges $\{(\pi(k), \pi(i)) = e_0, (\pi(j), \pi(q)) = e_1, (\pi(p), \pi(l)) = e_2\}$ of π with three other edges $\{(\pi(k), \pi(l)), (\pi(j), \pi(i)) = e, (\pi(p), \pi(q))\}$, which produce a modified permutation $\pi' = (\pi(1), \cdots, \pi(k), \pi(l), \cdots, \pi(j), \pi(i), \cdots, \pi(p), \pi(q), \cdots, \pi(n))$, see Fig.3(b). Since $d(\pi(k), \pi(i)) = d(\pi(j), \pi(q)) = d(\pi(p), \pi(l)) = 1$ and $d(\pi(j), \pi(i)) = \frac{1}{n^4}$, the cost of π' is better than that of the current path π. Similar to Case 1, we consider the transition probability that the ant switches from π to π'. Following Zhou[12], we need to implement 3-opt move which can obtain an improvement, and the probability of such an improvement is $\Omega(\frac{1}{n^5})$.

Altogether the theorem follows.

The theorem below shows that the (1+1) MMAA on Q_1 can reach the global optimal solution within expected polynomial time.

Theorem 3 Let $\alpha = 1$ and $\beta = 0$, $\tau_{\min} = \frac{1}{n^2}$ and $\tau_{\max} = 1 - \frac{1}{n}$. For arbitrary evaporation factor ρ with $0 < \rho \leq 1$, the expected optimization time of the (1+1) MMAA on Q_1 is $O(n^6 + \frac{n \log n}{\rho})$.

Proof Combing Lemma 4 and Theorem 2, we obtain the result.

V. Conclusions and Future Work

In this paper, we study the performances of ACO for the quadratic assignment problem. We reveal that the (1+1) MMAA can obtain some approximation solutions for the QAP problem. Furthermore, we reveal that ACO can defeat the local search algorithm with 2-exchange neighborhood on a QAP instance.

A natural question is whether the ACO algorithms can improve the approximation ratio which we have proven for the QAP problem. There is still much work to be done to understand well how the ACO algorithms work on optimization problems. For example, how the relationship of relative importance between pheromone trail and visibility affects the performance, though it may be an intractable and challenging problem.

References

[1] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, Cybernetics-Part B*, Vol.26, No.1, pp.29–41, 1996.

[2] M. Dorigo and T. Stützle, *Ant Colony Optimization*, Cambridge, MA: MIT Press, 2004.

[3] W.J. Gutjahr, "Mathematical runtime analysis of ACO algorithms: Survey on an emerging issue", *Swarm Intelligence*, Vol.1, No.1, pp.59–79, 2007.

[4] W.J. Gutjahr, "First steps to the runtime complexity analysis of ant colony optimization", *Computers and Operations Research*, Vol.35, No.9, pp.2711–2727, 2008.

[5] F. Neumann and C. Witt, "Runtime analysis of a simple ant colony optimization algorithm", *Algorithmica*, Vol.54, No.2, pp.243–255, 2009.

[6] S. Droste, T. Jansen and I. Wegener, "On the analysis of the (1+1) evolutionary algorithm", *Theoretical Computer Science*, Vol.276, No.1-2, pp.51–81, 2002.

[7] B. Doerr, F. Neumann, D. Sudholt, *et al.*, "Runtime analysis of the 1-ANT ant colony optimizer", *Theoretical Computer Science*, Vol.412, No.17, pp.1629–1644, 2011.

[8] T. Stützle and H.H. Hoos, "MAX-MIN ant system", *Future Generation Computer Systems*, Vol.16, No.8, pp.889–914, 2000.

[9] W.J. Gutjahr and G. Sebastiani, "Runtime analysis of ant colony optimization with best-so-far reinforcement", *Methodology and Computing in Applied Probability*, Vol.10, No.3, pp.409–433, 2008.

[10] F. Neumann, D. Sudholt and C. Witt, "Analysis of different MMAS ACO algorithms on unimodal functions and plateaus", *Swarm Intelligence*, Vol.3, No.1, pp.35–68, 2009.

[11] F. Neumann and C. Witt, "Ant colony optimization and the minimum spanning tree problem", *Theoretical Computer Science*, Vol.411, No.25, pp.2406–2413, 2010.

[12] Y. Zhou, "Runtime analysis of an ant colony optimization algorithm for TSP instances", *IEEE Transactions on Evolutionary Computation*, Vol.13, No.5, pp.1083–1092, 2009.

[13] T. Kötzing, F. Neumann, H. Röglin, *et al.*, "Theoretical analysis of two ACO approaches for the traveling salesman problem", *Swarm Intelligence*, Vol.6, No.1, pp.1–21, 2012.

[14] N. Attiratanasunthron and J. Fakcharoenphol, "A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs", *Information Processing Letters*, Vol.105, No.3, pp.88–92, 2008.

[15] D. Sudholt and C. Thyssen, "Running time analysis of ACO systems for shortest path problems", *Journal of Discreate Algorithm*, Vol.10, pp.165–180, 2012.

[16] A. Lissovoi and C. Witt, "MMAS versus population-based EA on a family of dynamic fitness functions", *Algorithmica*, Vol.75, No.3, pp. 554–576, 2016.

[17] T. Friedrich, T. Kötzing, M.S. Krejca, *et al.*, "Robustness of ant colony optimization to noise", *Evolutionary Computation*, Vol.24, No.2, pp.237–254, 2016.

[18] T.C. Koopmans and M.J. Beckman, "Assignment problems and the location of economic activities", *Econometrica*, Vol.25, No.1, pp. 53–76, 1957.

[19] S. Sahni and T. Gonzalez, "P-Complete approximation problems", *Journal of the ACM*, Vol.23, No.3, pp.555–565, 1976.

[20] E. Angel and V. Zissimopoulos, "On the quality of local search for the quadratic assignment problem", *Discrete Applied Mathematics*, Vol.82, No.1-3, pp.15–25, 1998.

[21] I. Wegener, "Methods for the analysis of evolutionary algorithms on pseudo-boolean functions", in: Sarker R, Mohammadian M and Yao X, eds. *Evolutionary Optimization*, Dordrecht: Kluwer, pp.349–369, 2002.

XIA Xiaoyun was born in 1982. He received the Ph.D. degree in computer software and theory from South China University of Technology. He is now a lecturer with the College of Mathematics, Physics and Information Engineering, Jiaxing University. His research interests include evolutionary computation, intelligent computation and global optimization. (Email: scutxxy@gmail.com)

ZHOU Yuren was born in 1965. He received the B.S. degree in mathematics from Peking University, Beijing, China, in 1988, the M.S. degree in the mathematics from Wuhan University, Wuhan, China, in 1991, and the Ph.D. degree in computer science from the same university in 2003. He is currently a professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests include design and analysis of algorithms, evolutionary computation. (Email: yrzhou@scut.edu.cn)