First Order Deceptive Problem of ACO and Its Performance Analysis

CHEN Ling
Department of Computer Science and Engineering, Yangzhou University , Yangzhou, China
State Key Lab of Novel Software Tech, Nanjing University， Nanjing, 210093, China
E-mail: lchen@yzcn.net

SUN Hai-Ying and WANG Su
Department of Computer Science and Engineering, Yangzhou University Yangzhou, China
Email: shy-eagle@163.com , yzwangshu@hotmail.com

Abstract—**Ant colony optimization(ACO), which is one of the intelligential optimization algorithm, has been widely used to solve combinational optimization problems. Deceptive problems have been considered difficult for ant colony optimization. It was believed that ACO will fail to converge to global optima of deceptive problems. This paper proves that the first order deceptive problem of ant colony algorithm satisfies value convergence under certain initial pheromone distribution, but does not satisfy solution convergence. We also present a first attempt towards the value-convergence time complexity analysis of ACO on the first-order deceptive systems taking the *n*-bit trap problem as the test instance. We prove that time complexity of MMAS, which is an ACO with limitations of the pheromone on each edge, on *n*-bit trap problem is O(n^2m.log *n*), here *n* is the size of the problem and *m* is the number of artificial ants. Our experimental results confirm the correctness of our analysis.**

Index Terms—**ant colony optimization, deceptive problems, *n*-bit trap problem**

I. INTRODUCTION

Ant colony optimization (ACO) [1-4] is a popular method for hard discrete optimization problems. In recent years, ACO has drawn much more attention of the researchers, and many improvements are advanced[5-10] . Due to its strong ability of optimization, ACO has been used to deal with practical applications such as traveling salesmen problem (TSP) [11], scheduling[12] , robot routing[13], data mining[14], machine learning and reasoning [15], network routing[16], wireless network configuring [17], frequency assignment [18], IIR filter design[19], quadratic assignment[20] , sequential ordering [21], weapon-target assignment[22] and other combinational optimization problems.

Although there are a huge amount of experimentations and variants of ACO, its theoretical foundation is still in its early development [2]. There are few theoretical studies for ACO, compared with the counterparts for genetic algorithm. Recently, there has been an increased effort to deepen the understanding of the convergence behavior of ACO. Gutjahr [23, 24] proves the convergence of a particular implementation of ACO called the graph-based ant system (GBAS). However, GBAS is quite different from common ACO implementations and its practical performance is unknown. Dorigo et al. shows the convergence of another class of ACO [1, 25], in which there is a lower bound τ_{min} to all pheromone values. Such a method is denoted as ACO$_{\tau min}$. Typically, there are two types of convergence of a stochastic optimization algorithm, i.e. convergence in solution and convergence in value. An algorithm has convergence in solution if $\lim_{t\to\infty} P_s(t)=1$, where $P_s(t)$ is the probability that the algorithm generates an optimal solution in the t-th iteration. An algorithm has convergence in value if $\lim_{t\to\infty} P_r(t)=1$, where $P_r(t)$ is the probability that the algorithm generates an optimal solution at least once in the 1st to tth iterations.Dorigo et al. [1,25] show that ACO$_{\tau min}$ achieves convergence in value under certain assumptions. Dorigo et al. [1, 25] also show the convergence in solution of ACO$_{\tau min}(t)$, in which the pheromone lower bound $\tau_{min}(t)$ changes over time, under the assumption that $\tau_{min}(t) = d/\ln(t+1)$, where d is a constant. The main limitation of studying ACO$_{\tau min}$ and ACO$_{\tau min}(t)$ is that they do not allow for an exponentially fast decrement of the pheromone trails resulted by using a constant evaporation factor, which is used by most ACO implementations.

A more important problem in the research of ACO is its speed of convergence, namely its runtime estimation. Runtime analysis of a metaheuristic algorithm has therefore to be done on a detailed level of the investigation of various specific test problems. In recent years, the analysis of metaheuristic algorithm with respect to their runtime become a growing research area and many results have been reported. Neumann and Witt are the first to analysis the runtime bounds of various ACO-based algorithms. In [26] they studied the behavior of the 1-ANT algorithm, and gave some probabilistic bounds for the 1-ANT on OneMax problem for small evaporation factor ρ . 1-ANT uses a single ant to construct solutions. If the number of variables in OneMax is n, they show with high probability that if $\rho = O(n^{-1-\varepsilon})$ for some $\varepsilon > 0$, the running time of the

doi:10.4304/jnw.4.10.993-1000

algorithm is $2^{\Omega(n^{\varepsilon/3})}$, and if for some $\rho = O(n^{-1+\varepsilon})$, the running time of the algorithm is $O(n^2)$. Neumann and Witt [26] also analyze the running time of the 1-ANT algorithm to construct minimum spanning trees. Another result on the runtime of a variant of ACO on the One-Max problem was reported by Gitjahr [27, 28]. They show that for values sufficiently close to one of the evaporation factor, the algorithm run time is of order $O(n\log n)$, which is the same order as that of (1+1)-Evolutionary algorithm. Nattapat Attiratanasunthron, Jittat Fakcharoenphol [29] prove polynomial running time bounds for an ACO algorithm for the single destination shortest path problem on directed acyclic graphs. They show that the expected number of iterations required for an ACO-based algorithm with n ants is $o\left(\frac{1}{\rho}n^2 m\log m\right)$ for graphs with n nodes and m edges, where ρ is an evaporation rate. This result can be modified to show that an ACO-based algorithm for One-Max with multiple ants converges in expected $o\left(\frac{1}{\rho}n^2\log m\right)$ iterations, where n is the number of variables. All these runtime estimations of ACO are based on a single ant version of ACO which are quite different from the classical ACO, and cannot reflect the performance of the ACO in real applications.

This paper proves that the first order deceptive problem of ant colony algorithm satisfies value convergence under certain initial pheromone distribution, but does not satisfy solution convergence. We also present a first attempt towards the value-convergence time complexity analysis of ACO on the first-order deceptive systems taking the n-bit trap problem as the test instance. We prove that time complexity of MMAS, which is an ACO with limitations of the pheromone on each edge, on n-bit trap problem is $O(n^2 m.\log n)$, here n is the size of the problem and m is the number of artificial ants. Our experimental results confirm the correctness of our analysis.

The rest of this paper is structured as follows: Section 2 introduces the ant colony optimization and its deceptive problems. In this section , we also describe the n-bit trap problem which is a first-order deceptive problem of ACO. Section 3 proves that the first order deceptive problem of ant colony algorithm satisfies value convergence under certain initial pheromone distribution. We also prove it does not satisfy solution convergence in Section 4. Section 5 analyzes the running time of MMAS on the n-bit trap problem. The experimental results are given in Section 6 to confirm our result of theoretical analysis. We finally conclude in Section 7with a summary.

II. Deceptive problems of ant colony optimization

In the ACO algorithm, we assign a value τ_i^j, called the pheromone, to each of the solution component c_i^j. The vector of all pheromone is denoted by τ.

The performance of an ACO algorithm can be evaluated by the expected iteration quality of the solutions generated in each iteration. The expected iteration quality is denoted as $W_F(\tau|t)$ where t is the iteration number. $W_F(\tau|t)$ is defined as:

$$W_F(\tau \mid t) = \sum_{s\in S} F(s)P(s \mid \tau) \qquad (1)$$

where $F{:}S \to \mathrm{R}^+$ is a quality function such that, for any s, $s' \in S$, if $f(s) > f(s')$, then $F(s) \geq F(s')$, $P(s|\tau)$ is the probability that the solution s is generated by an ant given the pheromone vector τ.

Definition 1 (local optimizer) Given a constrained optimization problem P, an ACO algorithm is a local optimizer for P if for any initial pheromone values, the expected iteration quality satisfies*:*

$$W_F(\tau \mid t+1) \geq W_F(\tau \mid t). \qquad (2)$$

We now introduce the notion of deception for local optimizers [30].

Definition 2 (first-order deceptive system) Given a constrained optimization problem P, it is called a first-order deceptive system (FODS) for an ACO algorithm, if the ACO algorithm is a local optimizer, and there exists an initial setting of pheromone values such that the algorithm does not in expectation converge to a global optimum.

This definition means that even if the model of ACO is a local optimizer, it is a first order deceptive system when it is applied to problem instances that are characterized by the fact that they induce more than one stable fix point of which at least one corresponds to a local optimum. Currently, whether a problem is a FODS is mostly proved by empirical studies, and lacks theoretical analysis. For example, experiments have been run to prove that the n-bit trap problem is a deceptive system by showing that the ACO does not converge to the global optimal solutions [30]. It was believed that the first order deceptive problem of ant colony algorithm satisfies value convergence under certain initial pheromone distribution, but does not satisfy solution convergence. This means even a problem, such like the n-bit trap problem, is a first-order deceptive system, it is possible for ACO to reach its optimum in the searching process. Further, understanding the time complexity for solving deceptive problems is important because that it sheds insights into the worst-case performance of ACO, as the deceptive problems are harder problems for ACO. Therefore, we study this issue on the n-bit trap problem, a well-known example of FODS for ACO, that has also been studied for evolutionary algorithms.

The n-bit trap problem is to find among the 2n binary numbers, from 0 to 2n-1, the one with the highest fitness. The fitness of a binary number s is defined as:

$$f(s) = \begin{cases} h(s) & s \neq 0 \\ n+1 & s = 0 \end{cases} \qquad (3)$$

where $h(s)$ is the Hamming distance between s and 0.

Obviously, the global optimum is $s^* = 0$. To solve the n-bit trap problem, we use m ants in the algorithm. In each iteration, an ant sequentially fixes the jth bit of the binary number in order of j = 1, 2,…,n. For the jth bit, the ant has two choices c_j^0 and c_j^1, corresponding to setting the *j*th bit to 0 and 1 respectively. The pheromone of c_j^0 and c_j^1 are τ_j^0 and τ_j^1 respectively. Define the set

$$G_j^k = \{(s^{(1)}s^{(2)}...s^{(n)}) \in S \mid s^{(j)} = k\} \quad (k = 0,1)$$

where $s^{(j)}$ is the value of the *jth* bit. Hence, G_j^k is the set of binary numbers whose *j*th bit is *k*. We define the function $F(G_j^k)$ as the summation of the fitness of all the numbers in G_j^k :

$$F(G_j^k) = \sum_{s \in G_j^k} f(s) \qquad (4)$$

For example, when n = 4, we have $F(G_j^0)$ =17 and $F(G_j^1)$ =20 for all *j*=1,2,3,4. For the *n*-bit trap problem and $F(G_j^k)$ function defined in (4), we have:

$$F(G_j^1) = (n+1)2^{n-2},\ F(G_j^0) = (n+1) + (n-1)2^{n-2}$$

and $$F(G_j^1) - F(G_j^0) = 2^{n-1} - n - 1.$$

We initialize pheromone τ_j^k as $\tau_j^k(0) = F(G_j^k)$ for all *j*=1,2,…,*n* and *k*=0,1. Since the values of $F(G_j^0)$ and $F(G_j^1)$ reflect the influence to the solutions when the ant selects 0 or 1 at the *j*th bit, we use them as the heuristic information. Therefore, we let $\eta_j^1 = 1$, $\eta_j^0 = \xi_0$, here ξ_0 is a constant satisfying $0<\xi_0<1$ and $F(G_j^0) = \xi_0 F(G_j^1)$ for a fixed *n*.

. For the *j*th bit, the probability an ant selects value c_j^k $(k = 0,1)$ is :

$$P(c_j^k, t) = \frac{\tau_j^k(t)\eta_j^k}{\tau_j^0(t)\xi_0 + \tau_j^1(t)} \qquad (5)$$

which is derived from (1) by setting $\alpha = \beta = 1$. In each iteration, the pheromone is updated as :

$$\tau_i^j(t+1) = \rho\tau_i^j(t) + \frac{1}{|S_i^j|}\sum_{s \in S_i^j} f(s) \qquad (6)$$

(j = 1,…*n*, *k*= 0,1) where ρ is the evaporation rate and S_i^j the set of solutions generated at the *t*th iteration that have c_i^j as the *i*th bit. By (6) , we know expected value of the pheromone can be obtained as follows:

$$E\left[\tau_i^j(t+1)\right] = \rho\tau_i^j(t) + \sum_{s \in G_i^j} f(s)P(s \mid \tau) \qquad (7)$$

III. SOLUTION CONVERGENCE OF ACO ON THE N-BIT TRAP PROBLEM

We name the ACO algorithm using (5) and (6) as selection and updating functions as ACO$_{n\text{-bit}}$. In this section we prove that the ACO$_{n\text{-bit}}$ algorithm can achieve value convergence but not solution convergence for a class of first order deceptive systems such as *n*-bit trap problem. First we give the following Lemma.

Lemma 1 Let γ >0 be a constant, we have

$$\lim_{T\to\infty}\prod_{t=\lceil\gamma\rceil+1}^{T}(1-\frac{\gamma}{t}) = 0$$

Then we present the following theorem to show that ACO$_{n\text{-bit}}$ can not achieve solution convergence.

Theorem 1 The ACO$_{n\text{-bit}}$ algorithm can not achieve solution convergence on the *n*-bit trap problem when *n*>3.

Proof: For the ACO$_{n\text{-bit}}$ algorithm, we define

$$\xi_j(t) = \frac{\tau_j^0(t)}{\tau_j^1(t)} \quad \forall j = 1,2,...,n,\ \ t = 1,2,....$$

Namely, $\xi_j(t)$ is the ratio of pheromone on value 0 to value 1 for bit *j* in iteration *t*.

For the *n*-trap problem, since the global optimal solution is s^*=（0, 0, ……, 0）, we show that for all *j*=1,2,…,*n*

$$\lim_{t\to\infty}E[\xi_j(t)] = \lim_{t\to\infty}E\left[\frac{\tau_j^0(t)}{\tau_j^1(t)}\right] = 0 \qquad (j=1,2,\cdots,\ n)$$

By (7), in the (*t*+1)th iteration, the expected value of the pheromone is :

$$E\left[\tau_i^j(t+1)\right] = \rho\tau_i^j(t) + \sum_{s \in G_i^j} f(s)P(s \mid \tau) \qquad (j = 0,1)$$

By（5）, the selection probabilities for c_j^0 and c_j^1 are $\frac{\xi_t\xi_0}{1+\xi_t\xi_0}$ and $\frac{1}{1+\xi_t\xi_0}$ respectively, then the expected value of the pheromone is :

$$E\left[\tau_j^0(t+1)\right] = \rho\tau_j^0(t) + \frac{\xi_t\xi_0}{1+\xi_t\xi_0}F(c_j^0),$$

$$E\left[\tau_j^1(t+1)\right] = \rho\tau_j^1(t) + \frac{1}{1+\xi_t\xi_0}F(c_j^1).$$

Since $$E\left[\frac{\tau_j^0(t+1)}{\tau_j^1(t+1)}\right] = \frac{\rho\xi_t\tau_j^1(t) + \frac{\xi_t\xi_0}{1+\xi_t\xi_0}F(G_j^0)}{\rho\tau_j^1(t) + \frac{1}{1+\xi_t\xi_0}F(G_j^1)}$$

$$= \frac{(1+\xi_t\xi_0)\rho\xi_t\tau_j^1(t) + \xi_t\xi_0 F(G_j^0)}{(1+\xi_t\xi_0)\rho\tau_j^1(t) + F(G_j^1)},$$

and $F(G_j^0) = \xi_0 F(G_j^1)$, where ξ_0 is a constant satisfying $0<\xi_0<1$, we have

$$E[\xi_{t+1}] = E\left[\frac{\tau_j^0(t+1)}{\tau_j^1(t+1)}\right] = \xi_t - \frac{(\xi_t - \xi_t\xi_0\xi_0)F(G_j^1)}{(1+\xi_t\xi_0)\rho\tau_j^1(t) + F(G_j^1)}$$

$$= \xi_t - \frac{\xi_t(1-\xi_0^2)F(G_j^1)}{(1+\xi_t\xi_0)\rho\tau_j^1(t) + F(G_j^1)} < \xi_t \qquad (8)$$

namely, $E(\xi_{t+1}) < E(\xi_t)$, which means

$$1 > \xi_0 > E(\xi_1) > E(\xi_2) > \cdots > E(\xi_t) > 0.$$

Since

$$\tau_j^1(t) = \rho\tau_j^1(t-1) + \frac{1}{1+\xi_{t-1}\xi_0}F_1(C_j^1) \le \tau_j^1(t-1) + F_1(C_j^1),$$

we have : $\tau_j^1(t) \le \tau_j^1(0) + tF_1(C_j^1)$.

Again since $\tau_j^1(0) = F_1(G_j^1)$, we can get $\tau_j^1(t) \le (t+1)F(G_j^1)$.

Because of $0 < \xi_t < 1$ and $0 < \xi_0 < 1$, we have $1 + \xi_t\xi_0 < 2$. By（8）, the expected value of the $\xi_j(t)$ satisfies

$$E\left[\frac{\tau_j^0(t+1)}{\tau_j^1(t+1)}\right] \le \xi_t\left[1 - \frac{1-\xi_0^2}{1+2\rho(t+1)}\right] \le \xi_t\left[1 - \frac{1-\xi_0^2}{(1+2\rho)(t+1)}\right].$$

Denote $\frac{1-\xi_0^2}{1+2\rho}$ as γ, then we have

$$E\left[\frac{\tau_j^0(t+1)}{\tau_j^1(t+1)}\right] \le \xi_t(1-\frac{\gamma}{t+1}).$$

Since $E\left[\frac{\tau_j^0(t+1)}{\tau_j^1(t+1)}\right] \le (1-\frac{\gamma}{t+1})E\left[\frac{\tau_j^0(t)}{\tau_j^1(t)}\right]$, we get

$$E\left[\frac{\tau_j^0(t)}{\tau_j^1(t)}\right] \le \prod_{k=1}^{t}(1-\frac{\gamma}{k})\cdot\frac{\tau_j^0(0)}{\tau_j^1(0)} = \prod_{k=1}^{t}(1-\frac{\gamma}{k})\cdot\xi_0$$

By Lemma 1, we have

$$\lim_{t\to\infty} E\left[\frac{\tau_j^0(t)}{\tau_j^1(t)}\right] = 0$$

Therefore, the ACO$_{n\text{-but}}$ algorithm will not converge to the optimal solution $s^* =$ (0, 0, ……, 0)

Q.E.D.

Theorem 1 gives an explanation why the ACO algorithm have difficulties in solving deceptive problems such as the *n*-bit trap problem.

IV. VALUE CONVERGENCE OF ACO ON THE N-BIT TRAP PROBLEM

Although ACO$_{n\text{-bit}}$ algorithm cannot achieve solution convergence, we will show that it can achieve value convergence. We need some preliminary results first.

Lemma 2 The pheromone in the *t*th iteration of ACO$_{n\text{-bit}}$ satisfies $\tau_j^0(t) \ge \frac{1-\rho^{t+1}}{1-\rho}\varphi F(G_j^0)$, where φ is a constant, 0< φ <1, for *j*=1,2,…,n and *t*=1,2,…

Proof : From the pheromone update rule in (6) , we have

$$\tau_j^0(t+1) = \rho\tau_j^0(t) + \frac{1}{|S_j^0|}\sum_{s\in S_j^0} f(s) \qquad (j = 1,\ldots n)$$

Here, S_j^0 is the set of all the solutions in current iteration with c_j^0 as the *i*th bit, obviously $S_j^0 \subseteq G_j^0$. Define $f_j^0 = \min\{f(s) \mid s \in S_j^0\}$, and $f_j^0 = \varphi F(G_j^0)$, clearly φ satisfies 0< φ <1. Since $\tau_j^0(t+1) \ge \rho\tau_j^0(t) + f_j^0$, we have

$$\tau_j^0(t+1) \ge \rho\tau_j^0(t) + \varphi F(G_j^0) \qquad (9)$$

By (9) we have

$$\tau_j^0(t) \ge \rho^t\tau_j^0(0) + (\rho^{t-1} + \rho^{t-2} + \ldots + 1)\varphi F(G_j^0)$$

Since $\tau_j^0(0) = F(G_j^0)$, we get

$$\tau_j^0(t) \ge \rho^t\varphi F(G_j^0) + (\rho^{t-1} + \rho^{t-2} + \ldots + 1)\varphi F(G_j^0)$$

$$= \frac{1-\rho^{t+1}}{1-\rho}\varphi F(G_j^0)$$

Q.E.D.

Lemma 3 For any integer *n* and a real number $\rho \in (0,1)$, there exists a positive integer t_0 such that $\frac{1-\rho^{t+1}}{1-\rho} > t^{\frac{-1}{n}}$ for any $t > t_0$.

Proof: We see the fact that

$$\frac{t^{\frac{-1}{n}}}{\frac{1-\rho^{t+1}}{1-\rho}} = \frac{1-\rho}{(1-\rho^{t+1})\sqrt[n]{t}} \ , \quad \text{and} \quad \lim_{t\to\infty}\frac{1-\rho}{(1-\rho^t)\sqrt[n]{t}} = 0.$$

Thus, there exists an integer t_0 >0, such that $\frac{t^{\frac{-1}{n}}}{\frac{1-\rho^{t+1}}{1-\rho}} < 1$ for $t > t_0$, which means $\frac{1-\rho^{t+1}}{1-\rho} > t^{\frac{-1}{n}}$.

Q.E.D.

From Lemma 2 and Lemma 3, we see that when *t* is large enough, $\tau_j^0(t) \ge t^{-\frac{1}{n}}\varphi F(G_j^0)$. We set $\tau_{\min}^0(j,t) = t^{-\frac{1}{n}}\varphi F(G_j^0)$. Since the value of $F(G_j^0)$ is independent of *j*, we can denote $F(G_j^0)$ as $F(G^0)$, and $\tau_{\min}^0(j,t)$ as $\tau_{\min}^0(t)$, namely

$$\tau_{\min}^0(t) = t^{-\frac{1}{n}}\varphi F(G^0) \qquad (10)$$

Lemma 4 The pheromone in the *t*th iteration of ACO$_{n\text{-bit}}$ satisfies $\tau_j^k(t) \le \frac{1}{1-\rho}F(G_j^k)$ (*j*=1,2,…,*n*, k =0,1) .

Proof: From the pheromone update rule in (6) , we have

$$\tau_j^k(t+1) = \rho\tau_j^k(t) + \frac{1}{|S_j^k|}\sum_{s\in S_j^k} f(s) \qquad (j = 1,\ldots n)$$

Since $S_j^k \subseteq G_j^k$, we have

$$\tau_j^k(t) \le \rho\tau_j^k(t-1) + F(G_j^k) \qquad (11)$$

And hence

$$\tau_j^k(t) \le \rho^t\tau_j^k(0) + (\rho^{t-1} + \rho^{t-2} + \cdots + 1)F(G_j^k)$$

Since $\tau_j^k(0) = F(G_j^k)$, we get

$$\tau_j^k(t) \le (\rho^t + \rho^{t-1} + \cdots + 1)F(G_j^k) = \frac{1-\rho^{t+1}}{1-\rho}F(G_j^k)$$

$$\le \frac{1}{1-\rho}F(G_j^k).$$

Q.E.D.

We define $\frac{1}{1-\rho}F(G_j^k) = \tau_{\max}^k(j)$ (*j*=1,2,…,*n*, *k*=0, 1) . Since the value of $F(G_j^k)$ is independent of *j*, we define $\tau_{\max}^k = \frac{1}{1-\rho}F(G^k)$, and $\tau_{\max} = \max\{\tau_{\max}^0, \tau_{\max}^1\}$.

Theorem 2 The ACO$_{n\text{-bit}}$ algorithm can achieve value convergence on the *n*-bit trap problem.

Proof : Although there are multiple ants, we only to prove that one ant can achieve value convergence in order to establish the result. Let $P(t)$ be the probability that the ant can find the optimal solution s^* in the *t*th iteration, and 1-$P(t)$ be the probability that the it can not find the optima in this iteration. From Lemma 3 and Lemma 4, we know that for $t>t_0$, where t_0 is defined in Lemma 3,

$$P(c_j^0,t)=\frac{\tau_j^0(t)\eta_j^0}{\tau_j^0(t)\eta_j^0+\tau_j^1(t)\eta_j^1}=\frac{\tau_k^0(t)\xi_0}{\tau_j^0(t)\xi_0+\tau_j^1(t)}$$
$$\geq\frac{\xi_0\tau_{\min}^0(t)}{(1+\xi_0)\tau_{\max}}.$$

Therefore we have $P(t)\geq\left[\frac{\xi_0\tau_{\min}^0(t)}{(1+\xi_0)\cdot\tau_{\max}}\right]^n$.

Let $P_{succ}(T)$ and $P_{fail}(T)$, respectively, be the probability that the any does and does not find s^* in the first T iterations, then we have $1-P_{succ(T)}=P_{fail}(T)=\prod_{t=1}^{T}(1-P(t))$. From (10) and Lemma 4 we know

$$P_{fail}(T)\leq\prod_{t=1}^{T}\left[1-\left(\frac{\xi_0[\tau_{\min}^0(t)]}{(1+\xi_0)\cdot\tau_{\max}}\right)^n\right]$$
$$\leq\prod_{t=1}^{T}\left[1-\left(\frac{\xi_0\varphi F(G^0)}{(1+\xi_0)\cdot\tau_{\max}}\right)^n\left(t^{-\frac{1}{n}}\right)^n\right].$$

Denote $\left[\frac{\xi_0\varphi F(G^0)}{(1+\xi_0)\cdot\tau_{\max}}\right]^n=\gamma$, then

$$\lim_{T\to\infty}P_{fail}(T)=\lim_{T\to\infty}\prod_{t=1}^{T}\left[1-\frac{\gamma}{t}\right].$$

From Lemma 1 we have that $\lim_{T\to\infty}P_{fail}(T)=0$, and thus $\lim_{T\to\infty}P_{succ}(T)=1$.

Q.E.D.

V. RUNTIME ESTIMATION

The deceptive problems of the ACO algorithm, such as the n-bit trap problem, are the most difficult problems for ACO. Although ACO can achieve value convergence in solving such problems, it costs large amount computation time to reach the global optimal solution. Therefore the time complexity for ACO to get the optima of a deceptive problem can be considered as the upper bound of ACO to solve arbitrary problems. In this section, we analyze the running time of MMAS（Max-Min Ant System）[1] on the n-bit trap problem. Stuezle T. et al [25] show that MMAS can achieve solution convergence.

In each iteration of the algorithm MMAS, the pheromone τ_j^k is updated as follows

$$\tau_i^k(t+1)=\max\left\{\rho\,\tau_i^k(t)+\Delta\tau\ ,\tau_{\min}\right\}\qquad(12)$$

Here, $0<b<1$ is a constant. We set the lower bound of the pheromone τ_j^k as $\tau_{\min}=\frac{cn}{n-c}$, and the evaporation factor as $\rho=\frac{c}{n}$, where $c>0$ is a constant. When selecting the component for variable X_i, the ant chooses c_i^j according to the probability function (5). In (5), we set $\eta_j^1=1$, $\eta_j^0=\xi_0$, here $\xi_0<1$ is a constant satisfying $F(G_j^0)=\xi_0F(G_j^1)$.

We denote the number of 0's in a solution s as $g(s)=n-h(s)$. We call the solution with the largest $g(s)$ value the algorithm obtained from the first to the tth iterations as the Hamming optimum solution at time t. Suppose after the tth iteration, the Hamming optimum solution is $y(t)$ where the number of 0's is $g(y(t))$, then $g(y(t))$ is a monotonic non-decreasing function of t. Suppose $t_{r+1}>t_r>0$, and $g(y(t_r-1))<r$, $g(y(t_r))=g(y(t_r+1))=g(y(t_r+2))=\ldots=g(y(t_{r+1}-1))=r$, $g(y(t_{r+1}))>r$, we call the period from the (t_r+1)th to t_{r+1}th iteration the rth stage of the algorithm. Let $t_0=0$ and $l_r=t_{r+1}-t_r$ be the number of iterations required by the rth stage. Then the time the algorithm required for reaching the optimum is

$$T=t_n=\sum_{r=0}^{n-1}(t_{r+1}-t_r)=\sum_{r=0}^{n-1}l_r\qquad(13)$$

For an integer $r\geq0$, denote the set of solutions with r bits of 0 as $S_r=\{s\mid s\in S,g(s)=r\}$, $r=0,1,2,\ldots,n-1$. In the rth stage, denote the time required for at least one ant to reach a solution $y'\in S_{r+1}$ as $T(S_r\to S_{r+1})$. Then $l_r=T(S_r\to S_{r+1})$. Suppose a solution $y=(y_1,y_2,\ldots,y_n)\in S_r$, define the set $S_{r+1}(y)=\{z=(z_1,z_2,\ldots,z_n)\mid\exists k(y_k=1,z_k=0,\ z_i=y_i,\text{ if }i\neq k)\}$, namely, the set consisting of the solutions of which all components are equal to the that of y except one bit which is 1 in y and 0 in z. Since $g(y)=r$, for every $z\in S_{r+1}(y)$, it satisfies $g(z)=r+1$, therefore we have $S_{r+1}(y)\subseteq S_{r+1}$. Suppose in the rth stage the time required for at least one ant to reach a solution $y'\in S_{r+1}(y)$ is $T(S_r\to S_{r+1}(y))$. Since $S_{r+1}(y)\subseteq S_{r+1}$, it is obvious that $T(S_r\to S_{r+1}(y))\geq T(S_r\to S_{r+1})$. Assuming an ant gets the global-best $y\in S_r$, denote the probability it reaches a solution $y'\in S_{r+1}(y)$ in one iteration as $P(y\to S_{r+1}(y))$, we have

$$T(y\to S_{r+1}(y))\geq T(S_r\to S_{r+1}(y))$$
$$\geq T(S_r\to S_{r+1}).\qquad(14)$$

Lemma 5 In solving the n-bit trap problem using MMAS, the expected value of $T(y\to S_{r+1}(y))$ satisfies

$$E[T(y\to S_{r+1}(y))]\leq\frac{2bn}{c\xi_0}.\frac{1}{n-r}\left(1-\frac{c}{(n-c)\xi_0}\right)^{1-n}\qquad(15)$$

Proof. Since $\tau_i^k(0)=bn$, $i=0,1$, $k=1,2,...,n$, by (12) it is easy to see that

$$\tau_i^k(t)\leq\rho\tau_i^k(t-1)+bn\leq\rho^2\tau_i^k(t-2)+\rho bn+bn$$
$$\leq\ldots\leq\rho^t\tau_i^k(0)+(\rho^{t-1}+\ldots+\rho+1)bn$$
$$=(\rho^t+\rho^{t-1}+\ldots+1)bn$$
$$\leq\frac{1}{1-\rho}bn=\frac{bn^2}{n-c}$$

Therefore, we denote the upper bound of pheromone τ_j^k as

$$\tau_{\max}=\frac{1}{1-\rho}bn=\frac{bn^2}{n-c}\qquad(16)$$

Then the probability for an ant to select c_j^k at the *j*th bit should satisfy

$$P_j^{(k)}(t)=\frac{\tau_j^k(t)\eta_j^k}{\tau_j^k(t)\eta_j^k+\tau_j^{1-k}(t)\eta_j^{1-k}}\ge\frac{\tau_{\min}\xi_0}{2\tau_{\max}}=\frac{\frac{cn}{n-c}\xi_0}{\frac{2bn^2}{n-c}}=\frac{c\xi_0}{2bn}. \quad (17)$$

We use $P_{\min}=\frac{c\xi_0}{2bn}$ to denote the minimum possibility for an ant to choose a component value at each bit. Assuming in the *t*th iteration, an ant selects c_j^k at the *j*th bit, then pheromone τ_j^k and τ_j^{1-k} are updated as follows .

$$\tau_j^k(t+1)=\rho\tau_j^k(t)+\Delta\tau\ge\rho\tau_{\min}+bn=\frac{\rho cn}{n-c}+bn=\frac{c^2}{n-c}+bn \quad (18)$$

$$\tau_j^{1-k}(t+1)=\rho\tau_j^{1-k}(t)\le\rho\tau_{\max}=\frac{\rho bn^2}{n-c}=\frac{bcn}{n-c} \quad (19)$$

Since $\tau_j^{1-k}(t+1)=\rho\tau^{1-k}(t)\ge\rho\cdot\tau_{\min}=\frac{\rho cn}{n-c}=\frac{c^2}{n-c}$, the probability for this ant to choose c_j^{1-k} in the $(t+1)$th iteration is:

$$p_j^{(1-k)}(t+1)=\frac{\tau_j^{1-k}(t+1)\cdot\eta_j^{1-k}}{\tau_j^{1-k}(t+1)\cdot\eta_j^{1-k}+\tau_j^k(t+1)\cdot\eta_j^k}$$
$$\le\frac{\frac{bcn}{n-c}}{\frac{c^2}{n-c}+\frac{c^2\xi_0}{n-c}+bn\xi_0}=\frac{\frac{bcn}{n-c}}{\frac{c^2(1+\xi_0)}{n-c}+bn\xi_0}$$
$$\le\frac{bcn}{bn(n-c)\xi_0}=\frac{c}{(n-c)\xi_0} \quad (20)$$

In the $(t+1)$th iteration, the probability for this ant to keep c_j^k unchanged is:

$$P_j^{(k)}(t+1)=1-P_j^{(1-k)}(t+1)\ge1-\frac{c}{(n-c)\xi_0} \quad (21)$$

Suppose an ant gets a solution $y\in S_r$, denote the probability it reaches a solution $y'\in S_{r+1}(y)$ in one iteration as $P(y\to S_{r+1}(y))$. Since the solution $y\in S_r$, it has *r* bits of 0 and *n-r* bits of 1. Denote the probability this ant gets a solution *y'* by shifting exactly one bit of 1 into 0 in one iteration as $P_a(y\to S_{r+1}(y))$. It is obvious that

$$P_a(y\to S_{r+1}(y))<P(y\to S_{r+1}(y))$$

and

$$P_a(y\to S_{r+1}(y))\ge P_{\min}(n-r)\left(1-\frac{c}{(n-c)\xi_0}\right)^{n-1} \quad (22)$$

Here, P_{min} is the lower bound of the probability for one bit shifting from 1 to 0. By（17）, we let $P_{\min}=\frac{c\xi_0}{2bn}$. Since *y* has *n-r* bits equal to 1, we multiply it by *n-r*. $\left[1-\frac{c}{(n-c)\xi_0}\right]^{n-1}$ in (22) is the probability for the other *n*-1bits to keep their value in *y* unchanged. By (22) we have

$$P_a(y\to S_{r+1}(y))\ge\frac{c\xi_0}{2bn}(n-r)\left(1-\frac{c}{(n-c)\xi_0}\right)^{n-1} \quad (23)$$

And hence

$$P(y\to S_{r+1}(y))\ge\frac{c\xi_0}{2bn}(n-r)\left(1-\frac{c}{(n-c)\xi_0}\right)^{n-1} \quad (24)$$

Since $E[T(y\to S_{r+1}(y))]=[P(y\to S_{r+1}(y))]^{-1}$, we get

$$E[T(y\to S_{r+1}(y))]\le\frac{2bn}{c\xi_0}.\frac{1}{n-r}\left(1-\frac{c}{(n-c)\xi_0}\right)^{1-n}.$$

Q.E.D.

Since $T(y\to S_{r+1}(y))\ge T(S_r\to S_{r+1}(y))\ge T(S_r\to S_{r+1})$, by (14), we can easily get the following lemma from Lemma 5.

Lemma 6 In solving the *n*-bit trap problem using MMAS, the expected value of $T(S_r\to S_{r+1})$ satisfies

$$E[T(S_r\to S_{r+1})]\le\frac{2bn}{c\xi_0}.\frac{1}{n-r}\left(1-\frac{c}{(n-c)\xi_0}\right)^{1-n} \quad (25)$$

Theorem 3. In solving the *n*-bit trap problem using MMAS, denote the number of iterations required to reach the optimal solution as $T(n)$, then $T(n)=O(n.\log n)$.

Proof. From Lemma 6 we know $E[T(S_r\to S_{r+1})]\le\frac{2bn}{c\xi_0}.\frac{1}{n-r}\left(1-\frac{c}{(n-c)\xi_0}\right)^{1-n}$. Since $l_r=T(S_r\to S_{r+1})$, the expected time length of the *r*th stage satisfies

$$E[l_r]\le\frac{2bn}{c\xi_0}.\frac{1}{n-r}\frac{1}{\left(1-\frac{c}{(n-c)\xi_0}\right)^{n-1}}$$

In the worst case, the initial global-best solution is $(1,1,\ldots,1)\in S_0$, and it is modified into $(0,0,\ldots,0)\in S_n$ through *n* stages. Then the total number of iterations required is the summation of the time of the *n* stages. That is

$$E[T(n)]=\sum_{r=0}^{n-1}E[l_r]=$$
$$\sum_{r=0}^{n-1}\left(\frac{2bn}{c\xi_0}.\frac{1}{n-r}.\left[1-\frac{c}{(n-c)\xi_0}\right]^{-(n-1)}\right)$$
$$=\frac{2bn}{c\xi_0}\left[1-\frac{c}{(n-c)\xi_0}\right]^{-(n-1)}\sum_{r=1}^{n}\frac{1}{r}\le\frac{2bn}{c\xi_0}e^{\frac{c}{\xi_0}}\sum_{r=1}^{n}\frac{1}{r}.$$

Since $\sum_{r=1}^{n}\frac{1}{r}=O(\log n)$, we get $T(n)=O(n.\log n)$。

Q.E.D.

From Theorem 3 we know that in solving the n-bit trap problem using MMAS, number of iterations required to reach the optimal solution is of order $O(n.\log n)$. Since in each iteration, every ant should select values of n bits, and on each bit there are two possible values, time complexity for each iteration is O(*nm*), here m is the number of ants in the system. Therefore upper bound of the time complexity of the algorithm is O($n^2 m.\log n$)。

VI. Experimental Results

We implement the MMAS on the *n*-bit trap problem on Pentium IV, Windows XP, P1.7G, using V C++ 6. 0, and visualize the results on Matlab 6.0.

We set n=10, evaporation rate ρ = 0.95, and use 20 ants. First, we test the convergence in value of the algorithm on *n*-bit trap problem. We make 100 runs and record the incumbent solutions. We plot in Figure 1 the average fitness of the incumbent solution f versus the iteration number t. We see that the average f approaches the optimal value 11 as t increases, showing the convergence in value of ACO.

To estimate the number of iterations of the algorithm to reaching the optimum of *n*-bit trap problem , we make runs on different n values from 5 to 25. We set the number of ants m=20, parameters b=0.95 and c=0.95n. Denote the number of iterations for reaching the optimum for *n*-bit as $T(n)$.

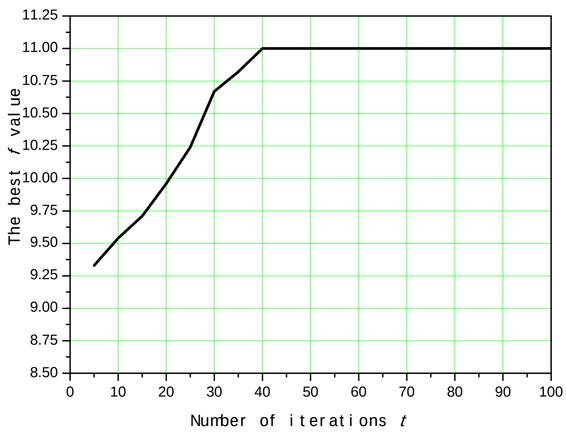

Figure 1 Average incumbent solutions

We plot in Figure 2 $T(n)$ and 0.95$n\log n$ versus different n values. From Figure 2 we can see that the curves of $T(n)$ approximate that of 0.95$n.\log n$ very closely, indicating the correctness of our conclusion by theoretical analysis.

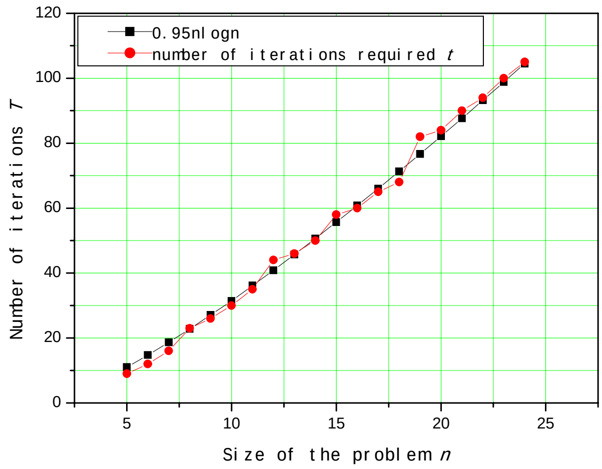

Figure 2 The curves of $T(n)$ approximate that of 0.95$n.\log n$ very closely

VII. Conclusions

Deceptive problems have been considered difficult for ant colony optimization . It was believed that ACO will fail to converge to global optima of deceptive problems. This paper proves that the first order deceptive problem of ant colony algorithm satisfies value convergence under certain initial pheromone distribution, but does not satisfy solution convergence. We also present a first attempt towards the value-convergence time complexity analysis of ACO on the first-order deceptive systems taking the n-bit trap problem as the test instance. We prove that time complexity of MMAS, which is an ACO with limitations of the pheromone on each edge, on *n*-bit trap problem is O($n^2 m.\log n$), here n is the size of the problem and m is the number of artificial ants. Our experimental results confirm the correctness of our analysis.

Acknowledgment

This work was supported in part by the Chinese National Natural Science Foundation under grant No. 60673060 and 60773103, Natural Science Foundation of Jiangsu Province under contract BK2008206, and Natural Science Foundation of Education Department of Jiangsu Province under contract 08KJB520012. The authors also thank Dr. Yixin Chen of Washington University in St Louis for his valuable suggestions.

References

[1] Dorigo M, Stützle T, *Ant Colony Optimization[M]*, MIT Press, (2004)
[2] Dorigo M, Blum C, Ant colony optimization theory: A survey[J], Theoretical Computer Science, 344 (2005) 243-278
[3] Blum C, Ant colony optimization: Introduction and recent trends[J], Physics of Life Reviews, 2 (2005) 353-373
[4] Shtovba S, Ant Algorithms: Theory and Applications[J], Programming and Computer Software, 31(2005) 167-178

[5] Dréo J, Siarry P, Continuous interacting ant colony algorithm based on dense hierarchy[J]，Future Generation Computer Systems, 20 (2004) 841-856

[6] Elbeltagi E, Hegazy T, Grierson D, Comparison among five evolutionary-based optimization algorithms[J], Advanced Engineering Informatics, 19 (2005) 43-53

[7] Stefan J, Daniel M, Middendorf&ast, Martin, Hossam E, Hartmut S,On Enforced Convergence of ACO and its Implementation on the Reconfigurable Mesh Architecture Using Size Reduction Tasks[J], The Journal of Supercomputing, 26 (2003) 221-238

[8] Martin M, Frank R, Hartmut S., Multi colony ant Algorithms[J], Journal of Heuristics, 8 (2002) 305-320

[9] Jun Sun, Sheng-Wu Xiong, Fu-Ming Guo, A new pheromone updating strategy in ant colony optimization[C], Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference, 1(2004)620-625

[10] Coleman C.M, Rothwell E.J, Ross J.E, Investigation of Simulated annealing, ant-colony optimization, and genetic algorithms for self-structuring antenna[J], IEEE Transactions on Antennas and Propagation, 52(2004)1007-1014

[11] Agarwal A., Lim M. H., Er M. J., Chew C. Y., ACO for a new TSP in region coverage[C], Proceedings of Intelligent Robots and Systems 2005, 2-6 Aug. (2005) 1717 – 1722

[12] Blum C, Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling[J], Computers & Operations Research, 32(2005) 1565-1591

[13] Prokopenko M, Peter Wang, Foreman M, Valencia P, Price D, Poulton G, On connectivity of reconfigurable impact networks in ageless aerospace vehicles, Robotics and Autonomous Systems, 53(2005)36-58

[14] Yixin, Chen; Ling, Chen; Li, Tu , Parallel ant colony algorithm for mining classification rules[C], Proceedings of 2006 IEEE International Conference on Granular Computing, (2006), 85-90

[15] R.J. Kuo, Y.P. Kuo, Kai-Ying Chen, Developing a diagnostic system through integration of fuzzy case-based reasoning and fuzzy ant colony system[J], Expert Systems with Applications, 28(2005)783-797

[16] Katangur, A.K., Akkaladevi S., Pan, Y., Fraser, M.D., Applying ant colony optimization to routing in optical multistage interconnection networks with limited crosstalk[C], Proceedings of 18th International Parallel and Distributed Processing Symposium, 2004., 26-30 (2004)163

[17] Shyong Jian Shyu, B.M.T. Lin , Tsung-Shen Hsiao, Ant colony optimization for the cell assignment problem in PCS networks[J] , Computers & Operations Research, 33(2006)1713-1740

[18] Maniezzo V., Carbonaro A., An ANTS heuristic for the frequency assignment problem[J] , Future Generation Computer Systems, 16(2000)927 – 935

[19] Kalinli A., Karaboga N., Artificial immune algorithm for IIR filter design[J] , Engineering Applications of Artificial Intelligence, 18(2005)919-929

[20] Talbi E.-G., Roux O., Fonlupt C., Robillard D., Parallel Ant Colonies for the quadratic assignment problem[J] , Future Generation Computer Systems, 17(2001)441-449

[21] Gambardella L. M., Dorigo M.,Ant Colony System hybridized with a new local search for the sequential ordering problem[J] , INFORMS Journal on Computing, 12(2000)237–255

[22] Lee Z.J., Lee C.Y., Su S.F, An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem[J] , Applied Soft Computing, 2(2002)39-47

[23] Gutjahr W.J, A Graph-based Ant System and its convergence[J] , Future Generation Computer Systems, 16(2000)873-888

[24] Gutjahr W. J., ACO algorithms with guaranteed convergence to the optimal solution[J] , Info. Processing Lett., 82(2002)145–153

[25] Stützle T., Dorigo M., A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms[J] , IEEE Transactions on Evolutionary Computation, 6,(2002) 358-365

[26] F. Neumann and C. Witt, Runtime analysis of a simple ant colony optimization algorithm, *Proceedings of ISAAC2006*, LNCS 4288, pp. 618-627

[27] W.J Gutjahr, On the finite-time dynamics of ant colony optimization, *Methodology and Computing in Applied Probability,* vol 8, pp. 105-133, 2006

[28] W.J Gutjahr, First step to the runtime complexity analysis of ant colony optimization, *Computers and Operations Research,* 2007

[29] Nattapat Attiratanasunthron, Jittat Fakcharoenphol, A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs, *Information Processing Letters,* 105(2008), pp88-92

[30] Blum C., Dorigo M., Deception in ant colony optimization[C], Proceedings of ANTS2004, Lecture Notes in Computer Science, 3172(2004)119-130

[31] Blum C., Dorigo M., Search bias in ant colony optimization: on the role of competition-balanced systems[J] , IEEE Transactions on Evolutionary Computation, 9(2005) 159 – 174

[32] Blum C., Dorigo M., The hypercube framework for ant colony optimization[J] , IEEE Transaction on Systems, Man, and Cybernetics—Part B, 43(2004)1161-1172

[33] Blum C., Theoretical and practical aspects of ant colony optimization[J] , Dissertations in Artificial Intelligence, Vol.2463, Akademische Verlagsgesellschaft Aka CmbH, Berlin, Germany, 2004

[34] Attiratanasunthron N., Fakcharoenphol J., A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs[J] , Information Processing Letters, V105, pp88-92, 2008

[35] Hao Z F, Huang H, Zhang XL, Tu K. A time complexity analysis of ACO for linear functions[C]. Lecture Notes in Computer Science，V4247. 513–520.

[36] Yixin Chen，Haiying Sun, Convergence of ant colony optimization on first order deceptive systems[C], Proceedings of 2008 IEEE International Conference on Granular Computing, pp158-163