INFORMATION
TECHNOLOGY JOURNAL

Convergence and Runtime of an Ant Colony Optimization Model

X. Yu and T. Zhang
School of Computer Science and Technology, Harbin Institute of Technology, 150001, China

Abstract: This study considered a model of an Ant Colony Optimization (ACO) algorithm for the general combinatorial optimization problem. The model proved that it can converge to one of the optima if only this optimum is allowed to update the pheromone model and that it can not converge to any of the optima if two or more optima are allowed. The iteration complexity of the model can be computed easily. And then a lower bound of time complexity of a real ACO algorithm for the general combinatorial optimization problem can be obtained.

Key words: Ant colony optimization model, combinatorial optimization problem, computational complexity

INTRODUCTION

The theory of Ant Colony Optimization (ACO) (Dorigo and Caro, 1999) is recently being built due to the success of applying ACO to several hard combinatorial optimization problems (Dorigo and Blum, 2005). The first theoretical issue is whether an ACO algorithm can converge to one of the optima of an instance of the problem being solved given sufficient computing resources. This is an interesting question because an ACO algorithm may not be able to converge to any optimum due to the pheromone model bias. Gutjahr (2000, 2002, 2003) earliest used the theory of Markov process to model and prove that the so called Graph-Based Ant System (GBAS) can converge probabilistically under given conditions to one of the optima. Stützle and Dorigo (2002) used the simple probability theory to prove that $ACO_{bs, \, min}$ with a lower limit of pheromone values can be probabilistically guaranteed to find an optimal solution (convergence in value) as well as $ACO_{bs, \, min \, (t)}$ under the condition that the bound τmin decreases to zero slowly enough and that a sufficiently slow decrement of the lower pheromone value limits results to the effect that the $ACO_{bs, \, min \, (t)}$ converges to a state in which all the ants construct the optimal solution repeatedly. Badr and Fahmy (2004) developed a proof of convergence for ant algorithms with the branching random walk and branching Wiener process. Baojiang and Shiyong (2006) and Haibin and Xiefen (2006) used Markov chains or martingale theory to suggest two proofs of ant algorithms. Zuo and Xiong (2006) proposed an improved version of ant algorithm and given the proof of its convergence. The convergence speed of an ACO algorithm is closely related to its convergence. However, the convergence results and their proofs state nothing about the convergence speed.

Determining the convergence speed of an ACO algorithm or more generally a meta-heuristic algorithm seems much harder. Gutjahr (2008a, b) discussed the runtime of GBAS and suggested a formula of an upper bound. Boumaza and Bruno (2008) discussed the convergence and rate of convergence of an ant foraging model which is a simple deterministic dynamic system. Another research line of the theory of ACO understands the behavior of ACO models. Blum and Dorigo (2004, 2005) reported the expected iteration quality of an ant system model gradually increases over time and analyzed the negative effectiveness of the pheromone model bias. Merkle and Middendorf (2002) modeled the dynamics of the ACO meta-heuristic and analyzed the so-called fix-point bias. Additionally, Gutjahr (2006) analyzed the finite-time dynamics of ACO.

According to literatures, no research considered the convergence and its speed of ACO models to this day. This research aimed to study the convergence and time complexity of an ACO model. First, the definition of the general combinatorial optimization problem was given and a simple ACO algorithm for it was defined. Then based on the model of this algorithm, three convergence theorems were justified. Further, the iteration complexity of this model was achieved and a lower bound of the time complexity of this algorithm was obtained. This research was closely related to open problem 1 and open problem 4 by Dorigo and Blum (2005).

ALGORITHM AND MODEL

The models analyzed in this study should be based on the algorithm which can solve the general combinatorial optimization problem with tiny modification and should be able to carry out theoretical analysis. To

---

this end, a general description of static combinatorial optimization problems was first presented. And then some basic concepts were given such as solution component, pheromone and search space.

General description of combinatorial optimization problem: The following is a formal characterization, as given for example by Blum and Dorigo (2004), of a combinatorial optimization problem.

Definition 1: A combinatorial optimization problem P = (S, f, Ω) is defined by:

- A set of discrete variables X_i with values $x_i \in D_i = \{d_j^i, ..., d_{|D_i|}^i\}, i = 1, ..., n$
- A set Ω of constraints among variables
- An objective function of f: $D_1 \times ... \times D_n \rightarrow R$ to be minimized

A solution is a row vector expressed as:

$$\bar{s} \overset{def}{=} b_1^i ... b_{D_1}^i ... b_1^n ... b_{D_n}^n, b_j^i \in \{0,1\}, \sum_{j=1}^{D_i} b_j^i = 1, 1 \leq i \leq n, 1 \leq j \leq D_i$$

which is a binary string and which length is

$$|D_1| \times |D_2| \times ... \times |D_n| . b_j^i = 1$$

is equivalent to

$$x_i \neq d_j^i \in D_i \text{ and } b_j^i = 0$$

equivalent to $x_i \neq d_j^i \in D_i$. Each solution $\bar{s}$ meets all the constraints. The collection of all possible feasible solutions is expressed as $S = \{\bar{s}\}$. The collection is often referred as a search (or solution) space, for each element of the collection can be considered as a candidate solution. If a solution $\bar{s}^* \in S$ meets that $f(\bar{s}^*) \leq f(\bar{s}) \forall \bar{s} \in S$, then $\bar{s}$ is referred as a globally optimal solution. The collection of all globally optimal solutions is expressed as $S^* \subset S$. Solving a combinatorial optimization problem is searching an optimal solution $\bar{s}^* \in S^*$ in its solution space.

Based on the above definition, a solution component is referred as b_j^i corresponding to the domain value of the variable X_i. Here the definition of solution component is different from that by Blum and Dorigo (2004) where, a solution component refers to the combination of the variable and one of its domain values. A partial solution is expressed as $\bar{s}^p$ (corresponding to a state of the problem) and the fact that a solution component b_j^i is part of the partial solution $\bar{s}^p$ expressed as $b_j^i \in \bar{s}^p$. It should be noted that if $b_j^i \in S^p, b_j^i, ..., b_{j-1}^i, b_j^i, b_{j+1}^i, b_{|D_i|}^i$ must be in $\bar{s}^p$.

The solution component b_j^i is associated with a pheromone value τ_j^i. The collection of all pheromone values is expressed as $\bar{\tau}$, that is, $\bar{\tau} = \{\tau_j^i : 1 \leq i \leq n; 1 \leq j \leq D_i\}$.

Algorithm: Algorithm 1 gives the high level description of applying ACO for the general combinatorial optimization problem described above, shortly AntCO. The pheromone model $\bar{\tau}$ saves the experience of the ant colony search which solution components should be set to 1 to obtain the best solution. The terminate condition generally refers to the maximum CPU time. AntCO is an iterative procedure consisting of two main sub-procedures in italic type. One is construct and the other update.

Algoirthm 1 AntCO

Initialize all pheromone values to 1.0;
Initialize setting all solution components to 0;
while Terminate condition is not met do
for each ant k = 1,...,m do
Construct a solution $\bar{s}^k$ according to (1);
end for
Find the best ant;
Update the pheromone model according to (2) by $\bar{s}^{ib}$;
end while

Solution construction: When starting to build a solution, ant k considers sequentially each variable, X_i, i = 1,..., n, randomly selects a value from the domain of the variable according to Eq. 1 and sets the solution component b_j^i to Eq. 1 and other solution components associated with the variable X_i to 0:

$$p(d_j^i | \bar{s}^p) = \begin{cases} \dfrac{\tau_j^i}{\sum_{j \in J(\bar{s}^p)} \tau_j^i} & \text{if } j \, \varepsilon \, J(S) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where, $J(\bar{s}^p)$ refers to the collection of the index of the domain values of the variable X_i to meet all the constraints; $p(d_j^i | \bar{s}^p)$ is probability selecting the domain value for the d_j^i variable X_i in the condition of the current partial solution $\bar{s}^p$.

Pheromone update: After all ants complete their solutions, one of the best ants $\bar{s}^{ib}$ at current iteration is allowed to update the pheromone model $\bar{\tau}$ as the following:

$$\tau_j^i \leftarrow (1-\rho)\tau_j^i + \rho\tau_j^i \quad (2)$$

where, $\rho \in (0, 1)]$ is the pheromone evaporating ratio; First $\rho\tau_j^i$ is subtracted from each pheromone value τ_j^i. If

$\bar{s}^{ib}$ set b_j^i to 1, ρ is added to τ_j^i. In this way, the probabilities of setting the variables, which are set to 1 by ant $\bar{s}^{ib}$, to 1 by the next ants increase and that of setting other variables to 0 increase.

Convergence factor: AntCO is convergent if all ants construct the same solution. AntCO is convergent to the optimum if this solution just is the global optimal solution. Otherwise, AntCO gets stagnant. One can decide whether AntCO is convergent according to the convergence factor:

$$cf = \frac{\sum_{i=1}^{n} \sum_{j=1}^{|D_i|} \tau_j^i}{n} \qquad (3)$$

At the beginning of the algorithm, because all pheromone values are set to 1.0, $cf = \sum_{i=1}^{n} |D_i| / n$ and setting each solution component to 1 and 0 has the same probabilities; the convergence factor cf gradually decreases during a run of AntCO; when the algorithm is convergent, each τ_j^i has the value very close to 1 or 0 and cf is very close to 1.

MODELS AND CONVERGENCE

A model of an ACO algorithm refers to the modified algorithm that the pheromone model update is replaced by an expected update rule (Blum and Dorigo, 2004) so that the iterative procedure become a deterministic dynamical system. To reduce the computational complexity one may consider $m = \infty$, that is, assume an infinite number of ants per iteration. For a model of AntCO considers an infinite number of ants per iteration, all solutions are constructed and those best solutions at each iteration must be all the optimums, that is, S*. Any instance of some combinatorial optimization may have only one, or two or two more optimal solution. If an instance has two or two more optimal solution, then one may update the pheromone model as the followings: (a) only use a specific optimal solution $\bar{s}^* \in S^*$, for example, use $\bar{s}^*$ at iteration 1, use $\bar{s}^*$ at iteration 2 and so on; (b) use two or two more optimal solutions at any iteration; (c) use one optimum at each iteration but two different optimal solution in two adjacent iterative step, for example, $\bar{s}^1 \in S^*$ at iteration t, $\bar{s}^2 \in S^*$ at iteration t+1, $\bar{s}^1 \in S^*$ at iteration t+3 and so on. Let M (G, IB, $m = \infty$) be a model of AntCO where, G represents the general combinatorial optimization problem described above, IB the method (a) of updating the pheromone model, m the number of ants. As M (G, IB, $m = \infty$). Similarly, Let M (G, IBS, $m = \infty$) and M (G, PIB, $m = \infty$) be another two models of AntCO where, IBS and PIB are the method (b) and (c) of updating the pheromone model,

respectively. The following theorems give the convergence of the three models.

Theorem 1: Assume $\bar{s}^g$ be one optimal solution of an instance of some combinatorial optimization problem being solved by AntCO and $\bar{s}^g$ always be used to update the pheromone model $\bar{\tau}$ at each iteration, M(G, IB, $m = \infty$), can be sure to convergent to the optimum $\bar{s}^g$.

Proof: Only need to justify that the pheromone values corresponding to the solution components set to 1 by the optimum $\bar{s}^g$ will be very close to 1 and the others very close to 0. Without loss of generalization, assume that $\bar{s}^g$ consists of n 1 characters $\sum_{i=1}^{n} |D_i| - n$ '0' and 0 characters. The pheromone model $\bar{\tau}(t)$ at iteration can be determined as the following rule:

$$\bar{\tau}(t) \leftarrow \bar{\tau}(t-1)A + \bar{b}$$

where, $\bar{\tau}(t)$ is a row vector consisting of all pheromone values $\tau_j^i(t)$ at iteration t:

$$A = \begin{bmatrix} (1-\rho)s_1^i & & & \\ & \ddots & & \\ & & (1-\rho)s_j^i & \\ & & & \ddots \\ & & & & (1-\rho)s_{|D_n|}^n \end{bmatrix}$$

is a $\sum_{i=1}^{n} |D_i| \times \sum_{i=1}^{n} |D_i|$ diagonal matrix which each element on the diagonal is $(1-\rho)s_j^i$; $s_j^i \in \{0,1\}$ is set the solution component b_j^i to by the optimum $\bar{s}^p$; $\bar{b}$ is also a row vector consisting of $\sum_{i=1}^{n} |D_i|$ elements of ρs_j^i. $\bar{\tau}(0)$ is a row vector consisting $\sum_{i=1}^{n} |D_i|$ of elements of 1.0. $\bar{\tau}(t)$ can be expressed $\bar{\tau}(0)$ and $\bar{b}$ by and as the following:

$$\bar{\tau}(t) = \bar{\tau}(0)A^t + \bar{b}(A^{t-1} + A^{t-2} + ... + A + A^0)$$

Clearly, $\lim_{t \to \infty} \bar{\tau}(t)$ consists of $\sum_{i=1}^{n} |D_i|$ elements of $1*s_j^i$, which is equal to the optimum $\bar{x}^g$. According to Eq. 1, all ants must construct the optimum.

If all optimal solutions S* can be allowed, the pheromone model rule is the following instead of Eq. 2 so that the pheromone values can not be greater than 1:

$$\tau_j^i \leftarrow \begin{cases} (1-\rho)\tau_j^i + \rho \sum_{\bar{s} \in S^*} \dfrac{s_j^i}{\bar{s} \in S^* s_j^i}, & \text{if } \sum_{\bar{s} \in S^*} s_j^i > 1, \\ (1-\rho)\tau_j^i, & \text{othwiseer} \end{cases} \qquad (4)$$

where, $\bar{s}$ sets the solution component b_j^i to $s_j^i \in \{0,1\}$. This formula states that if two (or more) different iteration-best solutions $\bar{s}^1$ and $\bar{s}^2$ and both set the solution component

b_j^i to 1, then the sum of the amount of pheromone value added to τ_j^i equals ρ to; if only one iteration-best solution $\bar{s}$ sets the variable b_j^i to 1 then ρ is added to τ_j^i.

Theorem 2: Assume $\bar{s}^1, \bar{s}^2$ be two different optimal solutions of an instance of some combinatorial optimization problem being solved by AntCO and $\bar{s}^1, \bar{s}^2$ always be used to update the pheromone model $\bar{\tau}$ at each iteration, M (G, IBS, m = ∞) can generally not converge to either of both optima except that there are only two different solution components in $\bar{s}^1, \bar{s}^2$ which are associated with the same variable.

Proof: Consider an instance having just two optima $\bar{s}^1$ and $\bar{s}^2$ such as...0(1)...(r)0..., where (i) denotes that the ith element is unset. For $\bar{s}^1, (l)^1 = 1$ and $(r)^1 = 0$ while $(l)^2 = 0$ and $(r)^2 = 1$ for $\bar{s}^2$. Assume that M (G, IBS, m = ∞) can find both optima $\bar{s}^1$ and $\bar{s}^2$ at any iteration, according to Eq. 4 and following the proof of Theorem 1, $\lim_{t \to \infty} \bar{\tau}(t)$ consists of $1.0 * \max[s_j^i(1), s_j^i(2)]$ where $(l) = 1$ $(r) = 1$. According to Eq. 1, if the two solution components are associated with the same variable, then an ant must construct one of two optima $\bar{s}^1$ and $\bar{s}^2$ and the probability of constructing them both 0.5. If the two solution components are associated with two different variables, then an ant construct one of two optima $\bar{s}^1$ and $\bar{s}^2$ with probability of 0.25. This is contrary to the assumed result. Clearly, we can easily obtain the similar result when an instance has more than two optima.

Theorem 3: Assume $\bar{s}^1, \bar{s}^2$ be two different optimal solutions of an instance of some combinatorial optimization problem being solved by AntCO and $\bar{s}^1, \bar{s}^2$ in turn be used to update the pheromone model $\bar{\tau}$ at each iteration, generally M (G, PIB, m = ∞) can not converge except that there are only two different solution components in $\bar{s}^1, \bar{s}^2$ which are associated with the same variable.

Proof: Consider two optima $\bar{s}^1$ and $\bar{s}^2$ as described in the proof of Theorem 2. Similarly assume that M (G, PIB, m = ∞) can find both optima $\bar{s}^1$ and $\bar{s}^2$ at any iteration. Let $\tau_l(t)$ and $\tau_r(t)$ correspond the elements (l) and (r). Suppose that the solution sequence allowed to is $\bar{s}^1, \bar{s}^2, \bar{s}^1, \bar{s}^2, ..., \bar{s}^1, \bar{s}^2$. According to Eq. 2:

$$\tau_l(t) \leftarrow \begin{cases} (1-\rho)^\tau \tau_l(0) + \dfrac{1-(1-\rho)^\tau}{2-\rho} & \text{if } t = 1(\text{mod } 2) \\[2ex] (1-\rho)^\tau \tau_l(0) + \dfrac{(1-\rho)(1-\rho)^\tau}{2-\rho} & \text{otherwise} \end{cases}$$

and

$$\tau_r(t) \leftarrow \begin{cases} (1-\rho)^\tau \tau_r(0) + \dfrac{(1-\rho)(1-\rho)^\tau}{2-\rho} & \text{if } t = 1(\text{mod } 2) \\[2ex] (1-\rho)^\tau \tau_r(0) + \dfrac{1-(1-\rho)^\tau}{2-\rho} & \text{otherwise} \end{cases}$$

Let k be an integer. If $k \to \infty$ and $t = 2k+1$, $\tau_l(t) \to \rho(1-\rho)/(2-\rho)$ and $\tau_r(t) \to (1-\rho)/(2-\rho)$. According to Eq. 1, if the two different solution components positioned at (l), (r) are associated with the same variable, then any ant construct $\bar{s}^1$ with probability $1/(2-\rho)$ and $\bar{s}^2$ with probability $(1-\rho)/(2-\rho)$, that is, M(g, PIB, m = ∞)can converge to the optimal value $f(\bar{s}^1) = f(\bar{s}^2)$. However, if the two different solution components positioned at (l), (r) are associated with the two different variables, then any ant construct $\bar{s}^1$ with probability $p(\bar{s}^1) = [1/(3-\rho)/(2-\rho)][(3-2\rho)]$ and $\bar{s}^2$ with probability $p(\bar{s}^2) = [(2-\rho)/(3-\rho)][(1-\rho)/(3-2\rho)]$. If $k \to \infty$ and $t = 2k$, similar results can be achieved easily. Generally, $\rho \in (0, 1)$. If $\rho \to 1$, $p(\bar{s}^1) \to 0.25$ and $p(\bar{s}^2) \to 0$. If $\rho \to 0$, $p(\bar{s}^1) \to 2/9$ and $p(vecs^2) \to 2/9$. In other words, M (G, PIB, m = ∞) can not converge to any of the optimal solutions.

In fact, the search of ACO algorithms in the solution spaces of combinatorial optimization problems is a gradual increasing learning process: learning which solution components can be used to build the optimal solution. This learning is based on the search history of ACO algorithms. This requires that ACO algorithms firstly explore fully the solution spaces of combinatorial optimization problems and then exploit the knowledge learn to guide the search concentrated into the hopeful sub space. Whenever a better solution is found, the search gradually converges to this new one. And this process is the same to a run of the model M (G, IB, m = ∞). Theorems 2 and 3 state that why one can only adopt the pheromone model update rule according to Eq. 2. And they can also explain the effectiveness of the update schema of the pheromone model in MMAS (Stützle and Hoos, 2000): during the early of a run, allowing iteration-best solution to update the pheromone model can aid in diversifying the search because iteration-best solution may vary from iteration to iteration; during the middle, allowing iteration-best and restart-best solutions to do can help in guiding the search towards to the hopeful area, at the same time avoiding the search getting stuck; during the late period, allowing so-far-best solution can make the search to converge to the optimum. In addition, they can also explain the invalidity of the method of updating the pheromone model in AS (Dorigo *et al.*, 1996): at each iteration, AS allows all ants to update the pheromone model and ants can not gradually converge to a better solution found. So, the pheromone values present the ant colony to move to the regional of obtaining the optimal solution.

LOWER BOUND OF TIME COMPLEXITY

How long $M(G, IB, m = \infty)$ takes to converge to the optimum of an instance is a very interesting and important question. Based on the answer one can achieve a lower bound of the time complexity of AntCO.

Theorem 4: The iteration complexity of $M(G, IB, m = \infty)$ for solving the general combinatorial optimization with n variables is $O(n)$.

Proof: Let ε be a very small positive real number. If the probability of constructing the optimum of an instance being solved by any ant is greater than $1-\varepsilon$, then $M(G, IB, m = \infty)$ can be said to reach convergence state from a view point of practice. This implies that for the general combinatorial optimization problem with n variables, the pheromone values associated with the variable X_i meet:

$$\frac{\tau_j^i}{\tau_j^i + \sum_{h=1 \text{ and } h \neq j}^{|D_i|} \tau_j^i} = \sqrt[n]{1-\varepsilon}$$

where, τ_j^i corresponds the solution component b_j^i set to 1 by the optimum, τ_h^i corresponds to the solution component $b_h^i, h \neq j, h \in \{1, ..., |D_i|\}$ set to 0. Considering, $\tau_j^i \rightarrow (1-\rho)^t \tau_j^i(0) + \rho \frac{1-(1-\rho)^t}{\rho}, \tau_h^i \rightarrow (1-\rho)^t \tau_h^i(0)$ and $\tau_j^i(0), \tau_h^i(0) = 1$, further the following formula can be obtained:

$$\frac{1}{1 + (|D_i|-1)(1-\rho)^t} = \sqrt[n]{1-\varepsilon}$$

When t is large enough, $(|D_i|-1)(1-\rho)^t$ will get very small and we have:

$$\frac{1}{1 + (|D_i|-1)(1-\rho)^t} = 1 - (|D_i|-1)(1-\rho)^t$$

So, we can obtain:

$$1 - (|D_i|-1)(1-\rho)^t = \sqrt[n]{1-\varepsilon}$$

When n is large enough, $1 - \sqrt[n]{1-\varepsilon} = \varepsilon^n$. If logarithm operation is carried out on the above equation, then we have:

$$t = \frac{n \log \varepsilon - \log(|D_i|-1)}{\log(1-\rho)}$$

As $\log \varepsilon$, $\log(|D_i|-1)$ and $\log(1-\rho)$ are constant, the iteration complexity of $M(G, IB, m = \infty)$ is $O(n)$.

Theorem 5: A lower bound of the time complexity of AntCO is $O(n \sum_{i=1}^{n} D_i |)$.

Proof: Generally, in most real ACO algorithms the number m of ants is constant and $m \ll n$. In AntCO shown as Algorithm 1, the time complexity of a WHILE loop is $O(\sum_{i=1}^{n} D_i |)$. Suppose that $m \rightarrow \infty$, then the iteration complexity of AntCO is the same as Theorem 4. So, a lower bound of the time complexity of AntCO is $O(n \sum_{i=1}^{n} D_i |)$.

CONCLUSION

If an Ant Colony Optimization (ACO) algorithm with infinite ants can not converge to any of the optima of an instance of the combinatorial optimization problem being solved, then one dare not expect this algorithm with very finite ants to find one of the optima. So, studying the convergence of an ACO algorithm model, which the number of ants is infinite, is of great importance. This study first designed an ACO algorithm for the general combinatorial optimization problem, shortly AntCO. And the convergence of the models of AntCO allowing different number of ants to update the pheromone model was discussed: if only one of the optima is allowed, the AntCO model can be sure to converge to the optimal solution; otherwise it must not converge to any of the optima. Additionally, the iteration complexity of the AntCO model is $O(n \sum_{i=1}^{n} D_i |)$ where, n is the number of variables and D_i is the value domain of variable $i = 1, ..., n$ and the time complexity of a real ACO algorithm for any combinatorial optimization problem is not less than $O(n \sum_{i=1}^{n} D_i |)$.

REFERENCES

Badr, A. and A. Fahmy, 2004. A proof of convergence for ant algorithms. Inform. Sci., 160: 267-279.

Baojiang, Z. and L. Shiyong, 2006. A convergence proof for ant colony algorithm. In: Jian, S. T.J. Tarn and C. Tianyou (Eds.). Proceeding of 6th World Congress on Intelligent Control and Automation, 2006 IEEE Piscataway, NJ., USA., pp: 3073-3075.

Blum, C. and M. Dorigo, 2004. The hyper-cube framework for ant colony optimization. IEEE Trans. Syst. Man Cybernet. B, 34: 1161-1172.

Blum, C. and M. Dorigo, 2005. Search bias in ant colony optimization: On the Role of competition-balanced systems. IEEE Trans. Evolu. Comput., 9: 159-174.

Boumaza, A. and S. Bruno, 2008. Convergence and rate of convergence of a foraging ant model. Proceeding of IEEE Congress on Evolutionary Computation, September 25-28, IEEE Piscataway, NJ., USA., pp: 469-476.

Dorigo, M., V. Maniezzo and A. Colorni, 1996. Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybernet. B, 26: 29-41.

Dorigo, M. and G. Di Caro, 1999. The Ant Colony Optimization Meta-Heuristic. In: New Ideas in Optimization, Corne, D., M. Dorigo and F. Glover (Eds.). McGraw Hill, New York, ISBN: 0077095065, pp: 11-32.

Dorigo, M. and C. Blum, 2005. Ant colony optimization theory: A survey. Theor. Comput. Sci., 344: 243-278.

Gutjahr, W.J., 2000. A graph-based ant system and its convergence. Future Gener Comp Sy, 16: 873-888.

Gutjahr, W.J., 2002. ACO algorithms with guaranteed convergence to the optimal solution. Inform. Proc. Lett., 82: 145-153.

Gutjahr, W.J., 2003. A generalized convergence result for the graph-based ant system. Probab. Eng. Inform. Sci., 17: 545-569.

Gutjahr, W.J., 2006. On the finite-time dynamics of ant colony optimization. Methodol. Comput. Applied, 8: 105-133.

Gutjahr, W.J., 2008a. First steps to the runtime complexity analysis of ant colony optimization. Comput. Oper. Res., 35: 2711-2727.

Gutjahr, W.J., 2008b. Runtime analysis of ant colony optimization with best-so-far reinforcement. Methodol. Comput. Applied, 10: 409-433.

Haibin, D., W. Daobo and Y. Xiufen, 2006. Markov chains and martingale theory based convergence proof of ant colony algorithm and its simulation platform. In: Jian, S., T.J. Tarn and C. Tianyou (Eds.). Proceeding of 6th World Congress on Intelligent Control and Automation, June 21-23, IEEE Piscataway, NJ, USA., pp: 3057-3061.

Merkle, D. and M. Middendorf, 2002. Modeling the dynamics of ant colony optimization algorithms. Evolu. Comput., 10: 235-262.

Stützle, T. and H.H. Hoos, 2000. Max-min ant system. Future Gener Comp. Sy, 16: 889-914.

Stützle, T. and M. Dorigo, 2002. A short convergence proof for a class of ACO algorithms. IEEE Tranc. Evolu. Comput., 6: 358-365.

Zuo, H.H. and F.L. Xiong, 2006. A new approach of ant colony algorithm and its proof of convergence. In: Jian, S., T.J. Tarn and C. Tianyou (Eds.). Proceeding of 6th World Congress on Intelligent Control and Automation, June 21-23, IEEE Piscataway, NJ, USA., pp: 3301-3304.