Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 191 (2007) 42–56
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Ant colony pattern search algorithms for unconstrained and bound constrained optimization ☆

Y.J. Feng *, L. Yu, G.L. Zhang

College of Information Engineering, Zhejiang University of Technology, 310032 Hangzhou, China

Abstract

Ant colony optimization is a class of metaheuristics which succeed in NP-hard combinational optimization problems rather than continuous optimization problems. We present and analyze a class of ant colony algorithms for unconstrained and bound constrained optimization on R^n: Ant Colony Pattern Search Algorithms (APSAs). APSAs use the ant colony framework guided by objective function heuristic pheromone to perform local searches, whereas global search is handled by pattern search algorithms. The analysis results of APSAs prove that they have a probabilistic, weak stationary point convergence theory. APSAs present interesting emergent properties as it was shown through some analytical test functions.

Keywords: Ant colony optimization; Bound constrains; Stochastic pattern search; Convergence

1. Introduction

Ant colony optimization (ACO) is a cooperative search algorithm inspired by the behavior of real ants. One basic idea of the ACO approach is to use the counterpart of the pheromone trail used by real ants as a medium for communication and as an indirect form of memory of previously found solutions. The artificial ants are simple agents that have some basic capabilities, e.g. in the case of the traveling salesman problems (TSP) they may use some heuristic information like the distance between cities to construct tours and maintain a tabu list in which they store the partial tour constructed so far. In the past years, ant system has been applied in TSP problems [1], water distribution system optimization [2], project portfolio selection [3], minimum spanning tree problems [4], flow shop problems (FSP) [5,7], time series data mining [6] etc. Until now, there are few adaptations of such algorithms to continuous space function optimization problems. In this paper, we consider the application of ACO to solve unconstrained minimization problems

☆ This work was partially supported by the National Science Fund for Distinguished Youth Scholars of China Grant # 60525304 and the Natural Science Foundation Fund of Zhejiang Province China Grand # Y106660.
* Corresponding author.
E-mail addresses: fyjing@zjut.edu.en (Y.J. Feng), lyu@zjut.edu.cn (L. Yu), zgj@zjut.edu.cn (G.L. Zhang).

0096-3003/$ - see front matter © 2006 Published by Elsevier Inc.
doi:10.1016/j.amc.2006.09.142

$$\begin{aligned} &\text{Min} && f(x) \\ &\text{subject to} && x \in R^n, \end{aligned} \tag{1.1}$$

where $f : R^n \to R$, as well as bound constrained problems

$$\begin{aligned} &\text{Min} && f(x) \\ &\text{subject to} && x \in R^n, \quad l_i \leqslant x_i \leqslant u_i, \ \ i = 1, \ldots, n, \end{aligned} \tag{1.2}$$

where $l_i, u_i \in R$, and $l_i < u_i$. These problems have been solved using EAs like evolutionary programming (EP), genetic algorithm (GA), and ACOs like continuous ant colony optimization (CACO) [8,9], continuous interacting ant colony (CIAC) [10]. In this paper, we introduce and analyze ant colony pattern search algorithms (APSAs), a class of CACOs that can be used to solve problems (1.1) and (1.2). Like CACO and CIAC, APSAs simulate the real ants finding the food source.

A number of applications to many different hard discrete combinatorial optimization problems [1–5], has empirically shown the effectiveness of ant colony optimization. Still, very little theory is available to explain the reasons underlying ACO's success. Birattari et al. [11] have proposed an interpretation of ACO in the framework of optimal control and reinforcement learning, while Meuleau and Dorigo [12] have shown that ACO algorithms and stochastic gradient descent are strongly related and that a particular form of ACO algorithms converges with probability 1 to a local optimum. The work presented in Gutjahr's paper[13], a graph-based ant system was presented which tend with a probability $p \geqslant 1 - \varepsilon$ to the optimal solution. Closer to the work of Thomas and Dorigo [14], a short convergence of the ant system that applies directly to at least two of the most (experimentally) successful ACO algorithms: (1) ant colony system (ACS) [1] and (2) Max–Min ant system (MMAS) [15]. In our recent work [16], some convergence properties of a Metropolis Criterion base ant system were presented. But all the above works in convergence theory focused on the ACO in discrete optimization problems. For continuous domains, there are few results shown in the literatures.

Our analysis of APSAs provides the first stationary point convergence theory for adaptive continuous ant colony optimization on nonconvex, continuously differentiable functions. In particular, we show that for an unconstrained, continuously differentiable function, the sequence of the best individual found by and APSA, x_k^*, has the property that

$$P\{\liminf_{k\to\infty} \|g(x_k^*)\| = 0\} = 1, \tag{1.3}$$

where $g(x)$ is the gradient of f at x. This means that the set of sequence that do not converge has probability zero of occurring. Further, for a bound constrained, continuously differentiable function a subsequence of $\{x_k^*\}$ converges almost surely to a constrained stationary point (equivalently, a Karush–Kuhn–Tucker point for problem (1.2). These results can be extended to prove convergence almost surely on continuous nondifferentiable functions to limit points where the gradient does not exist or where the gradient is not continuous [17].

Further, this convergence theory exactly captures the convergence properties of APSAs. Our analysis of APSAs is based upon the analysis of stochastic pattern search algorithms in [18,19]. We generalize the analysis of stochastic pattern search to APSAs and demonstrate the correspondence between APSAs and stochastic pattern search. The results can be extended to other continuous ant colony optimization algorithms.

The organization of the remaining contents is as follows: In Section 2 the APSAs for unconstrained and bound constrained optimization is described. In Section 3 the probabilistic weak station-point convergence theory for APSAs is generalized. The computational results with typical unconstrained and bound constrained optimization problems are given and analyzed in Section 4. In Section 5 we offer some concluding remarks and points to ongoing work.

2. Ant colony pattern search algorithms

Ant colony optimization used in continuous space is different with that used in discrete space [20]. First, the trail deposit mode of the ant colony in each optimization circulation should not correspond to discrete points or discrete components. The trail left by the ants should also affects on the ambient field of these points. Thus the description of ant trail deposition should be in the form of distribution function, whose peak value corresponds to the value of objection function according to current ant system distribution state in the answer

space. Second, the optimization process of ant system in the answer space should not be the abrupt changing form between discrete answers. It should be in the differential transferring form. Third, because in resolution of the continuous space optimization problem, the trail deposition and affection mode is regional and not in point distribution, judging of ant colony movement should be based on the trail integration value comparing results in corresponding regions, not at discrete sets. Thus, the definition of continuous ACO should tackle trail pheromone distribution function, pheromone update strategy and the migration pattern which comprises local and global searches on the space. The first algorithm designed for continuous function optimization was CACO [8,9] which comprises two levels: global and local. CACO uses the ant colony framework to perform local searches, whereas global search is handled by a genetic algorithm. Indeed, the "global" ants perform a simple evaluation of some regions defined in the search space, in order to update the regions fitness. The creation of some new regions is handled by a process very similar to a genetic algorithm, using common operators that are assimilated by the authors to some real ant colonies behavior like "random walk" (playing the part of crossovers and mutations). The local level is handled by ants that explore more systematically the regions with a simple descending behavior, in order to move regions closer to the optimum. The algorithm sends some local ants on regions, these ants lay down some pheromonal spots when they find an improvement of the objective function and the spots are attractive for all the ants of the colony. The work of [10] presented a heterarchical algorithm called *Continuous interacting ant colony* (CIAC) which used tow communication channels showing the properties of trail and direct communications based on CACO. Closer to our work in [21], an ant system for continuous space multi-modal function optimization was proposed by introduced the immunity algorithm handling the global search. Unfortunately, the use of two different processes inside the CACO algorithm leads to a delicate setting of parameters.

The definition of APSAs also should tackle trail pheromone distribution function, pheromone update strategy and the migration pattern which comprises local and global searches on the space. After randomly distributing of the ants, each ant can deposit a certain amount of pheromone as a spot on the search space, proportionally to the improvement of the objective function she founds on her way. These pheromonal spots can be perceived by all the members of the population, and diffuse into the environment. The ants are attracted by each spot according to its distance and the amount of pheromone it contains, then making pattern search around the spot for further improving.

Individuals position initialized strategy and the pheromone distribution. First, we consider the bound constrained problems (1.2). From (1.2), there exits $l_i \leqslant x_i \leqslant u_i$, $i \in n$, in the ith dimension. We part the region $[l_i,u_i]$ into N parts, here N is the individual number, and take the center of N parts as the initial position of the individuals, see Fig. 2.1. So, it can be written as

$$x_{ij} = l_i + \frac{(2j-1)(u_i - l_i)}{2N}, \quad i \in n, \quad j \in N \tag{2.1}$$

For unconstrained problems (1.1), the position initialization can also be considered as constrained problems by randomly choosing the initial region $[l_{i1},u_{i1}]$.

The jth individual deposits initial equal pheromone value in the ith dimension. Then the pheromone updating strategy will be suggested in the next sections.

Ant migration pattern. The searching path of individual is shown as Fig. 2.1. The local search agents are moving according to the pheromonal spots cloud. An ant makes with a probability $q_0 \in [0,1]$ to select the best possible decision of arg max $([\tau_j]^{\alpha}[\eta_j]^{\beta})$,

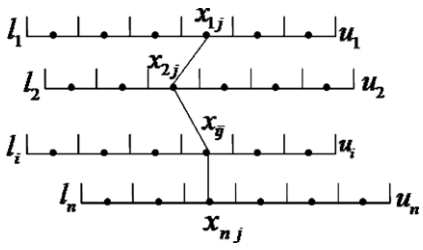

Fig. 2.1. The searching path of individuals.

$$s = \begin{cases} \arg\max([\tau_j]^{\alpha}[\eta_j]^{\beta}), & q \leqslant q_0, \\ s', & q > q_0, \end{cases} \tag{2.2}$$

and with probability $1 - q_0$ for the region s' chosen according to the following probability means the probability of ant i selecting region j.

$$p_{ij} = \frac{[\tau_j]^{\alpha}[\eta_j]^{\beta}}{\sum_{k \in J_{ck}}[\tau_k]^{\alpha}[\eta_k]^{\beta}}, \tag{2.3}$$

where η_j is the heuristic information. In this paper, if $f(x_j) \neq f(x^*), \eta_j = \min(1/|f(x_j) - f(f(x^*)|, \psi)$; else, $\eta_j = \psi$, means the gap to the minimum point x^*. $\psi > 0$ is a constant, J_{ck} is the feasible domain.

The exploratory move. After the ants finish migration, the exploratory move of each ant around the immigration region creates new regions by replacing the weaker portions of the existing domain. In APSA terminology these are called random walk and trail diffusion. By the random walk procedure, the ants move to new regions in search of newer and richer stocks of food sources.

The exploratory moves algorithm contains an iterative loop, indexed by h. In each iteration, the exploratory moves algorithm may randomly consider an exploratory move from a generating matrix C_i^k. The exploratory moves algorithm (shown in Fig. 2.2) take a trail step such that $s_i \in \{0, \pm 1, \pm h\}$; this offers a simple mechanism by which h can affect the set of possible trails steps. Note that the probability of sampling a coordinate vector e_i or $-e_i$ is greater than or equal to a constant $\gamma > 0$ independent of the value of h. Thus the vectors $\{\pm e_1, \ldots, \pm e_n\}$ represent a set of core exploratory moves for this algorithm.The algorithm may sample among the additional steps $\{0, \pm 1, \pm h\} \setminus \{\pm e_1, \ldots, \pm e_n\}$ in each iteration, which vary based on h.

Note. Let R, Q, Z and N denote the sets of real, rational, integer, and natural number, respectively.

Suppose $\phi \in Q$, such that $\phi > 1$ and $\phi = \phi_n/\phi_d$, where $\phi_n, \phi_d \in N$ and ϕ_n, ϕ_d are relatively prime. Let $\theta = \phi^{\varpi_0}$ and $\lambda_t \in \{\phi^{\varpi_1}, \ldots, \phi^{\varpi_L}\}$, where $\{\varpi_0, \ldots, \varpi_L\} \subset Z, L < \infty, \varpi_0 < 0$, and $\varpi_i \geqslant 0, i = 1, \ldots L$. The update algorithm of step size Δ_t is given by Fig. 2.3.

From the step size update algorithms, we can see that, after t iteration, the value of Δ_t is

$$\Delta_t = \Delta_0 \theta^{\alpha_0} \lambda_1^{\alpha_1} \ldots \lambda_L^{\alpha_L} = \Delta_0 \phi^{r_t} \tag{2.4}$$

where $\alpha_i \in Z$; the α_i are simply values that arise in the expression of Δ_t. Let $r^t_{\max} = \max_{i=1,\ldots,t} r_t$. Then define $\overline{\phi}_t = \phi_n^{-r^t_{\min}} \phi_d^{-r^t_{\max}}$.

Pheromone update. The division of G global ants deputed for searching by random pattern search is again an important variable in the APSA algorithm. Once G new regions are created, it is mandatory for us to set the

For $h = 1, 2, \cdots$
  Randomly select $s \in \{0, \pm 1, \pm h\}^n$
  If $(f(x_t + \Delta_t) < f(x_t))$ $s_t = \Delta_t s$
    break
  Else If $s \in \{0, \pm 1\}^n$
    If (*all vectors* $\pm e_i$ *have been generated*) $s_t = \{0\}^n$
      break
Endfor
$x_{t+1} = x_t + s_t$
Update Δ_t

Fig. 2.2. The exploratory move.

If $(f(x_t) \leq f(x_t + s_t))$
  $\Delta_{t+1} = \theta \Delta_t$
If $(f(x_t) > f(x_t + s_t)$ *and* s_t *is a core trial step.*)
  $\Delta_{t+1} = \lambda_t \Delta_t$

Fig. 2.3. Updating algorithm of Δ_t.

trail value of these new regions. It is a tough task to set the trail value of still unexplored regions; assigning the child region a trail value lying between the values of the original parent regions circumvents this difficulty. In real ant systems there is a process of food resource exhausting which causes the ants to lose interest. This feature is incorporated in the APSA algorithm by reducing the trail value of all regions by a certain value after each iteration step. With such a decrease the weaker regions are less likely to be selected as the simulation proceeds. The pheromone evaporation is carried out by algorithm of Fig. 2.4. In Fig. 2.4, $0 < \rho < 1$ is the evaporation rate, $\tau_{\min} > 0$ is a constant, and $h(x) < \infty$ is a function of x with: $f(x) < f(x') \Rightarrow h(x) \geqslant h(x')$.

The ant colony pattern search algorithm is generalized in Fig. 2.5 (the symbols used to describe this code are summarized in Table 2.1). An APSA is initialized with an initial step Δ_0 length. A finite set of matrices are chosen so that for all $\overline{S} \in S, \overline{S}$ is a rational matrix whose columns form a positive basis (so any point can be generated by a positive linear combination of the columns of S). The ants randomly select points in Ω and lay

$$
\begin{aligned}
&\forall k, \tau_k(t+1) \leftarrow (1-\rho)\tau_k(t)\\
&\textit{If } f(x_t) < f(x^*), \textit{ then } x^* \leftarrow x_t\\
&\forall k \in x^* : \tau_k(t+1) \leftarrow \tau_k(t+1) + \rho \cdot h(x_i)\\
&\forall k : \tau_k \leftarrow max\{\tau_{min}, \tau_k\}
\end{aligned}
$$

Fig. 2.4. Updating pheromone.

(1) *Initialize parameters.*
(2) *Select an initial population according to (2.1).*
(3) $x_0^* = argmin\{f(x_1^0), \cdots, f(x_N^0)\}$,*initialize pheromone value* τ_i^0.
(4) *Repeat* $t = 1, 2, \cdots$
(5) *For* $i = 1 : N$
(6) *For* $j = 1 : n$
(7) *If* $(unif() \leq q_0)$
(8) $\hat{x_{ij}} = argmax([\tau_j^t]^\alpha [\eta_j^t]^\beta)$,*according to (2.2).*
(9) *Else*
(10) $\hat{x_{ij}}$ *selected according to (2.3).*
(11) *For* $i = 1 : N$
(12) *If*$(unif() \leq \gamma)$
(13) $k = unit(m)$
(14) *If* $(\hat{x_i} + \Delta_t s_k$ *is not feasible*$)$ $\xi_k = 1$
(15) *Else If* $(\hat{x_i} == x_t^*)$ $\xi_k = 1$
(16) $\hat{x_i} = \hat{x_i} + \Delta_t s_k$
(17) $x_i^{t+1} = \hat{x_i}$
(18) $x_{t+1}^* = argmin\{f(x_1^{t+1}), \cdots, f(x_N^{t+1})\}$
(19) *If* $(f(x_{t+1}^*) < f(x_t^*))$
(20) *If* $(\exists s \in S$ *s.t.* $x_{t+1}^* = x_t^* + s)$ $\Delta_{t+1} = \lambda_t \Delta_t$.
(21) $\xi = \{0\}^m$
(22) *ElseIf* $(|\xi| == m)$
(23) $\xi = \{0\}^m$
(24) $\Delta_{t+1} = \theta \Delta_t$
(25) *Else*
(26) $\Delta_{t+1} = \Delta_t$
(27) *For bound constrained problems(1.2)*
(28) *If*$(\hat{x_{ij}} < l_i)$,$\hat{x_{ij}} = l_i$
(29) *If*$(\hat{x_{ij}} > u_i)$,$\hat{x_{ij}} = u_i$
(30) *For* $i = 1 : N$
(31) $\tau_i^{t+1} \leftarrow (1-\rho)\tau_i^t$
(32) *If* $(f(x_i^{t+1}) < f(x_t^*))$
(33) $\tau_i^{t+1} \leftarrow \tau_i^{t+1} + \rho \cdot h(x_i^{t+1})$
(34) $\tau_i^{t+1} \leftarrow max\{\tau_{min}, \tau_i^{t+1}\}$
(35) *Until the stopping criterion is satisfied.*

Fig. 2.5. The ant colony pattern search algorithm.

Table 2.1
Summary of the symbols used to describe APSAs

Symbol	Description
n	The dimension of the optimization function
N	The population size of ant colony
S	The trial step set
q_0	The constant with $0 < q_0 < 1$
ρ	The evaporation rate
Δ_t	The step size
γ	The constant with $0 < \gamma < 1$
$\hat{x}_i$	The position after ant migration pattern
unif()	A function that generates a uniformly distributed real in [0.0,1.0]
unit(m)	A function that generates a uniformly distributed integer in $1, \ldots, m$
x_t^*	The best objection function value in iteration t
ξ	The vector that records the trail steps that have been sampled
τ_i^t	The trail value of the ith region in iteration t
$\tau_{\min}$	The minimum value of the trail value
η_j^t	The heuristic information of the jth region
θ	The expansion factor
λ_t	The contraction factor

down some pheromonal spots round the points. In the optimization progress of APSA, the ants first make with the migration pattern as (2.2) and (2.3). Then, in each region,the ant takes the exploratory move to find the richer food source(the pattern search for better objective function value). In new region, the pheromone is recalculated and updated as Fig. 2.4.

3. Convergence theory

3.1. APSAs as stochastic pattern search

For convergence analysis, we cast the ant colony pattern search algorithm into the framework of the pattern search by Torczon [17] who defined pattern search in three central components: the generating matrix, the exploratory move, the step length update.

For APSA, the individuals generated at each generation are viewed as patterns with respect to the best individual in the population. The ant select the region according to (2.2) and (2.3) for stochastic pattern search. The generating matrix C_k^h is constructed using all possible individuals that can be generated in the current generation t. Let $f(x_1^t) \leqslant f(x_j^t), j = 1, 2, \ldots, N$. Then let $\kappa = lcd(1/\Delta_0, x_{k,j}^0, s_{i,j}, i = 1, 2, \ldots, m, k = 1, 2, \ldots, N, j = 1, 2, \ldots, n)$, which represents the least common denominator of the current points, the initial step length, and the pattern search vectors. The basis matrix implicitly used by ASPA is $B = \frac{1}{\kappa^3} I$, where is the identity matrix. The generating matrix $\Gamma_\kappa = \kappa^3[S_1, \ldots, s_m]$ define the stochastic pattern search around the best individual in each generate.

Before convergence analysis, we give several propositions of APSAs.

Proposition 3.1. *In ASPA, for any τ_k, the following holds*:

$$\lim_{n\to\infty} \tau_k(t) \leqslant \tau_{\max} = h(x^*). \tag{3.1}$$

Proof. According to the definition of $g(x)$, we know the maximum possible amount of pheromone added to any region k after any iteration is $h(x^*)$. Clearly,

$$\tau_k(1) \leqslant (1-\rho)\tau_0 + \rho \cdot h(x^*),$$
$$\tau_k(2) \leqslant (1-\rho)^2\tau_0 + (1-\rho)\rho \cdot h(x^*) + \rho \cdot h(x^*), \ldots,$$
$$\tau_k(t) \leqslant (1-\rho)^t\tau_0 + \rho \sum_{i=1}^{t} (1-\rho)^{t-i} h(x^*).$$

Then,

$$\lim_{n\to\infty}\tau_k(t)=\lim_{n\to\infty}(1-\rho)^t\tau_0+\lim_{n\to\infty}\left[\sum_{i=1}^{t}(1-\rho)^{t-i}\right]h(x^*). \tag{3.2}$$

In (3.2), asymptotically, as $0<\rho<1$, followings hold:

$$\lim_{n\to\infty}(1-\rho)^t\tau_0=0,\ \lim_{n\to\infty}\left[\sum_{i=1}^{t}(1-\rho)^{t-i}\right]h(x^*)=h(x^*).$$

So,

$$\lim_{n\to\infty}\tau_k(t)\leqslant\tau_{\max}=h(x^*). \qquad \square$$

Proposition 3.2. *For the sum of pheromone trails on solutions space, following hold*:

$$\lim_{n\to\infty}\sum\tau_k(t)\leqslant|v|h(x^*), \tag{3.3}$$

where $|v|$ is the number of pheromonal spots.

Proof. According to Proposition 3.1 there exists, $\lim_{n\to\infty}\sum\tau_k(t)=\sum\lim_{n\to\infty}\tau_k(t)\leqslant|v|h(x^*)$. $\square$

The step length updating algorithm of APSAs is given as Fig. 2.3. If an improving point is generated in current generation, then the step length may be increased. If all the offspring have been examined, then the step length is shrunk. The control of step length is realized by the restriction of parameter θ and λ_t.

The exploratory move algorithm of APSAs described in Fig. 3.1 contains the regions chosen and the exploratory move in Fig. 2.2, only terminates if a solution is found that generates a simple decrease of if all feasible steps defined by $\Delta_k B\Gamma_k$ have been examined.

3.2. Convergence analysis

Having set up the machinery to define pattern search methods, we are now ready to analyze these methods referencing to the theories of Torczon [17] and Hart [18,19]. This analysis will produce theorems of several types. The first, Lemma 3.3 and Lemma 3.4 demonstrate an algebraic fact about the nature of ant colony pattern search methods that requires no assumption on the function f. These theorems are critical to the proof of the convergence results for it shows that we only need require simple decrease in f to ensure global convergence. The second theory and the third theory developed in Lemma 3.5 guarantees that each of the exploratory moves terminate with probability one. The second theory, denoted as Lemma 3.6, describes the

Repeat $h=1,2,\cdots$
  For $i=1:N$
    $\hat{x_i}=selective(X_t)$
  For $i=1:N$
    If $(unif()\leq\gamma)$
      $j=unit(m)$
    If $(\hat{x_i}+\Delta_t s_j$ *is not feasible*$)$ $\xi_j=1$
    Else If $(\hat{x_i}==x_t^*)$ $\xi_j=1$
    $\hat{x_i}=\hat{x_i}+\Delta_t s_j$
    $x_i^{t+1}=\hat{x_i}$
    $x_{i+1}^*=argmin\{f(x_1^{t+1}),\cdots,f(x_N^{t+1})\}$
    $t=t+1$
Until $f(x_{t+1}^*)<f(x_t^*)$ *or* $(|\xi|==m)$

Fig. 3.1. The exploratory move algorithm of APSA.

limiting behavior of the step length control parameter Δ_k which approaches zero with probability one for the unconstrained and bound constrained cases. The main results are Theorem 3.7 and Theorem 3.8, which prove a stationary-point convergence for unconstrained and bound constrained APSAs, respectively.

Lemma 3.3. *There exists a constant $\delta > 0$, independent of k, such that for any core trial step $s_k^i \neq 0$ produced by a APSA the following holds*:

$$\|s_k^i\| \geqslant \delta \Delta_k. \tag{3.4}$$

Proof. $\|s_k^i\| = \|Bc_k^i\|\Delta_k \geqslant \sigma_{B\min}\|c_k^i\|\Delta_k \geqslant \sigma_{B\min}\Delta_k$ where $\sigma_{B\min}$ is the small est singular value of B, c_k^i is a nonzero integer, and hence $\|c_k^i\| \geqslant 1$. □

Lemma 3.4. *The best individual x_t^* generated by ant colony pattern search at any iteration can be expressed in the following form*:

$$x_t^* = x_0^* + \phi_n^{l_{\min}^k}\phi_d^{-r_{max}^k}\Delta_0 B\sum_{k=0}^{t-1} z_k, \tag{3.5}$$

where, $z_k \in Z^n, k = 0, 1, \ldots, t-1$.

$$r_{\max}^k = \max_{i=1,\ldots,k} r_k \quad \text{and} \quad r_{\min}^k = \min_{i=1,\ldots,k} r_k. \tag{3.6}$$

Proof. The ant colony pattern search algorithm guarantees that any iterate is of the form

$$x_t^* = x_0^* + \sum_{k=0}^{t-1} s_k = x_0^* + \sum_{k=0}^{t-1} \Delta_k B c_k.$$

Consider the step length parameters Δ_k which is updated according to Eq. (2.4), so we have

$$x_t^* = x_0^* + \Delta_0 B\sum_{k=0}^{t-1} \phi^{r_k} c_k.$$

Recall that $\bar{\phi}_t = \phi_n^{-r_{\min}^l}\phi_d^{-r_{\max}^l}$, where, ϕ_n, ϕ_d are relatively prime, so we have

$$x_t^* = x_0^* + \phi_n^{-r_{\min}^k}\phi_d^{-r_{\max}^k}\Delta_0 B\sum_{k=0}^{t-1} z_k$$

for $z_k \in Z^n$. □

Lemma 3.5. *Let A be the set of sequences of trial steps for which each exploratory terminates. Then,the terminating probability of each exploratory $P(A) = 1$.*

Proof. At iterate t, due to the pheromone trail $\tau_{\max}$ and $\tau_{\min}$, suppose the heuristic information $0 < \eta_{\min} \leqslant \eta_j \leqslant \psi$, that we can guarantee that the choice probability in (2.3) holds with following:

$$p_{i,j}(t) = \frac{\tau_j^\alpha \eta_j^\beta}{\sum_{l\in J_{ck}} \tau_l^\alpha \eta_l^\beta} \geqslant \frac{\tau_{\min}^\alpha \eta_{\min}^\beta}{(|v|-1)\tau_{\max}^\alpha \eta_{\max}^\beta + \tau_{\min}^\alpha \eta_{\min}^\beta} > \frac{\tau_{\min}^\alpha \eta_{\min}^\beta}{|v|[h(s^*)]^\alpha \psi^\beta}, \tag{3.7}$$

where J_{ck} is the feasible domain.

Suppose that, at each region, the probability of selecting each of the core steps is greater than or equal to a constant $\varphi > 0$. For APSAs, according to Eqs. (2.2) and (2.3), an ant selecting each core step with the

probability in two situation. An ant makes with the probability $\hat{p}(t) = q_0 \cdot \varphi > 0$ to select each core step in the region which satisfied arg max $([\tau_j^{\alpha}][\eta_j]^{\beta})$. And selecting others with the probability

$$\hat{p}(t) = (1 - q_0)\varphi p_{ij}(t) > (1 - q_0)\varphi \frac{\tau_m i n^{\alpha} \eta_{\min}^{\beta}}{|v|[h(s^*)]^{\alpha}\psi^{\beta}} > 0.$$

Recall that the exploratory moves algorithm terminates if a simple decrease is found or if all trial steps $\gamma_k = |v|, |\Gamma_k| \leqslant 2n|v|$ have been sampled. Let $R_{r,i,j}$ be the set of sequence of trial steps which the jth core step is not sampled in steps $r + 1$ through $r + i$, and let $R_{r,i}$ be the set of sequence of trial steps which one or more of the core steps is not sampled in steps $r + 1$ through $r + i$. Let R_r be the set of the sequence of trail steps which one or more of the core steps is not sampled in all of the trial steps following the rth trail step. Note that $R_{r,i} = \cup_{j=1}^{\gamma_k} R_{r,i,j}, R_r = \cap_{i=1}^{\infty} R_{r,i}$, and that $R_{r,i} \supset R_{r,i+1}$. We have

$$P(R_r) = \lim_{i\to\infty} P(R_r, i) \leqslant \lim_{i\to\infty} \sum_{j=1}^{\gamma_k} P(R_{r,i,j}) \leqslant \lim_{i\to\infty} \gamma_k (l - \hat{p})^i = 2n|v| \lim_{i\to\infty} (1 - \hat{p})^i = 0.$$

Thus we have $P(A) = 1 - P(\cup_{r=1}^{\infty} R_r) = 1 - \lim_{r\to\infty} P(R_r) = 1$. □

Lemma 3.6. *For unconstrained (bound constrained) ant colony pattern search, suppose that $L(x_0)(L_{\Omega}(x_0))$ is compact. Then $P(\liminf f_{k\to\infty} \Delta_k = 0) = 1$.*

Proof. The proof is analogous to the proofs of theorem 4 in [17] by contradiction. Suppose $0 < \Delta_{\min} \leqslant \Delta_k = \phi^{\gamma k}\Delta_0$ for all k. This means that the sequence ϕ^{γ_k} is bound away from zero. Meanwhile, we know that the sequence Δ_k is bounded above because all the iterates x_k must lie inside the set $L(x_0)(L_{\Omega}(x_0))$ which is compact; Lemma 1 then guarantees an upper bound $\Delta_{\max} < \infty$. From which it follows that the sequence ϕ^{γ_k} and r_k is bounded above and below. Let

$$r_{\max}^k = \max_{0\leqslant k<\infty} r_k \quad \text{and} \quad r_{\min}^k = \min_{0\leqslant k<\infty} r_k. \tag{3.8}$$

Then (3.5) now holds for the bounds given in (3.8), rather than (3.6), and x_k lies in the translated integer lattice generated by x_0 and the columns of $\phi_n^{r_{\min}^k}\phi_d^{-r_{\max}^k}\Delta_0 B$ for all k. □

The inter section of the compact set $L(x_0)(L_{\Omega}(x_0))$ with the translated integer lattice Φ is finite. Thus, there must exist at least one point x_* in the lattice for which $x_k = x_*$ for infinitely many k.

According to the step length updating strategy, we accept a new step s_k if and only if $f(x_t) > f(x_t + s_k)$, so there exists M such that for all $k \geqslant M, x_k = x_*$. This implies that $\rho_k = 0$ for $k \geqslant M$, but this implies that $\Delta_k \to 0$, which gives a contradiction to our assumption that $0 < \Delta_{\min} \leqslant \Delta_k = \phi^{\gamma_k}\Delta_0$. From Lemma 3.5 we know that the set of sequences for which each exploratory move terminates has measure one, so we have $P(\liminf_{k\to\infty} \Delta_k = 0) = 1$.

The previous results provide the basis for the convergence of unconstrained and bound constrained ant colony pattern search methods. Specifically, they guarantee that the sequence of iterates is implicitly constrained to a translated, scaled, and rotated integer lattice, and that the step size asymptotically convergence to zero almost surely. Then we will give the main results of this paper in the following context.

Let $\Omega = (x \in R^n | l \leqslant x \leqslant u)$. Define $L_{\Omega}(y) = (x \in \Omega | f(x) \leqslant f(y))$, for convenience,let $L(y) = L_{R^n}(y)$. And let

$$p_i(t) = \begin{cases} l_i, & \text{if } t < l_i, \\ t, & \text{if } l_i \leqslant t < u_i, \\ u_i, & \text{if } t > u_i. \end{cases} \tag{3.9}$$

and consider the projection of $x \in R^n$ onto the feasible region of problem (1.2),

$$Q(x) = \sum_{i=1}^{n} p_i(x_i) e_i. \tag{3.10}$$

where e_i is the ith standard basis vector. Note that x is a stationary point of (1.2) if and only if $q(x) = Q(x - g(x)) - x = 0$. In the bound constrained theory, the quantity $q(x)$ plays the role of the gradient

$g(x)$, providing a continuous measure of how close x is to a constrained stationary point. If $g(x) = 0$ such that the constraints are not active, then clearly $q(x) = Q(x - g(x)) - x = Q(x) - x = 0$. Otherwise, $q(x)$ is a continuous function that is zero at a point where $g(x)_i \leqslant 0$ if $x_i = u_i, g(x)_i \geqslant 0$ if $x_i = l_i$, and $g(x)_i = 0$ if $l_i < x_i < u_i$. Thus $q(x) = 0$ ensures that x is a constrained stationary point.

Theorem 3.7. *Let $L(x_0)$ be compact and suppose that $f : R^n \to R$ is continuously differentiate on an open neighborhood of $L(x_0)$. Then for the sequence of iterates produced by the unconstrained APSA,*

$$P\left(\liminf_{k\to\infty} \|g(x_k^*)\| = 0\right) = 1. \tag{3.11}$$

Theorem 3.8. *Let $L_\Omega(x_0)$ be compact and suppose that $f : R^n \to R$ is continuously differentiate on an open neighborhood of $L\Omega(x_0)$. Then for the sequence of iterates produced by the bound constrained APSA*

$$P\left(\liminf_{k\to\infty} \|q(x_k^*)\| = 0\right) = 1. \tag{3.12}$$

4. Experiment

To test the performances of our APSA optimization algorithms, several benchmark unconstrained test functions with different dimensions were selected. These typical benchmark functions considered in the literatures are shown in Appendix. In APSA, the pattern search method used is Hooke and Jeeves' pattern search algorithm which is essentially a variant of coordinate search that incorporates a pattern step in an attempt to accelerate the progress of the algorithm by exploiting information gained from the search during previous successful iterations. Torczon [17] had shown that the pattern search algorithm of Hooke and Jeeves is a special case of generalized pattern search method and so Theorem 3.7 and Theorem3.8 hold. The algorithm was written in MATLAB 6.0 program and run on a Pentium IV 2.4G PC with 256 RAM. To avoid any problem due to the choice of an initial population, we performed each test 20 times with different random seed at each test. The values of the objective function evaluation number (evals), the average CPU time(time) and of the average error (err) are obtained through averaging the results over all the tests. A test is considered to be successful (thus contributing to percent of ok, %) if Eq. (4.1) held in a given function evaluation number (here we given to 10,000):

$$|f^* - f^\alpha| < \varepsilon_1 f^* + \varepsilon_2, \tag{4.1}$$

where, f^* is the global optimum of the objective function, f^α is the optimum found by the algorithm, ε_1 and ε_2 are accuracy parameters: in all tests discussed in that paper, $\varepsilon_1 = \varepsilon_1 = 10^4$.

An algorithm is viewed as effectively, if it is good at three performances: reliable, scalable and flexible. Firstly, experiments were conducted with varying APSA parameters such as population size (N), step length control parameters (Δ_0), persistence coefficient (ρ). The average statical results of the computations with different test functions are shown in Table 4.1. Also, the comparison results of one varying parameters (the others are fixed) of N, Δ_0, ρ, with the Griewangk test function (the dimension of the function, $n = 5$) are shown in Figs. 4.1–4.3, respectively. From Fig. 4.1, we can found that the computation results is little dependent on the persistence coefficient (ρ), the optimization progress with large coefficient may cause more time consuming. Shown as Fig. 4.2, too small population size may lead to converge slowly due to the lack of communication information between individuals. In Fig. 4.3, the results with $\Delta_0 = 1$ shown that, in small initial trail step length case, the optimization progress converges slowly in initial iterations, but sharply converges to the optima later. The results with $\Delta_0 = 10$ shown that too large initial trail step length may cause undulation.

Further, the scalable performance APSA is tested with Rosenbrok and Griewangk functions in different scale of dimension. We set the parameters with population size N to 20, persistence coefficient ρ to 0.98, initial

Table 4.1
Comparison of APSA performance with different parameters

Fun. ($n = 5$)	Interval	N	ρ	Δ_0	Err	(%)	Time (s)	Evals
Rosenbrock	[−2.56 2.56]	20	0.98	5.0	7.6832e−6	100	94.2651	687
		20	0.9	1.0	9.6548e−5	100	69.3685	521
		10	0.9	10.0	6.9658e−5	99	69.5860	658
Griewangk	[−5.12 5.12]	10	0.7	10.0	9.6921e−5	96	32.1638	231
		20	0.9	5.0	9.2384e−5	100	26.5426	180
		20	0.98	1.0	9.4593e−4	69	186	1236
Rastrigrin	[−2 2]	10	0.7	5.0	2.4874	95	11.7173	54
		20	0.9	0.7	9.3625e−6	100	12.4376	16
		20	0.98	1.0	9.6542e−6	100	22.3765	28
Sphere model	[−50 50]	10	0.7	5.0	7.3691e−7	100	2.6954	26
		20	0.9	1.0	6.9869e−6	100	25.5861	56
		30	0.98	10	2.8481e−6	100	11.6272	11

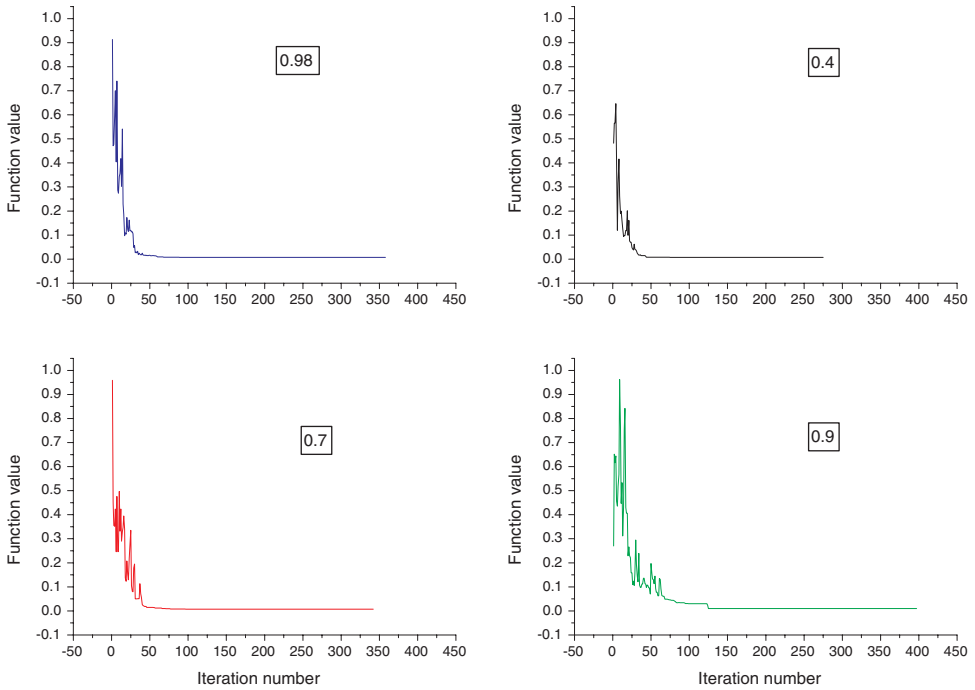

Fig. 4.1. Comparison results of APSA with different ρ.

step Δ_0 to 1.0. The results shown in Table 4.2 suggest that APSA not only do well in little scale of dimension problems but also deal with the large scale of dimension ones well.

Next, the performances of the APSA are compared with other continuous ant colony optimization such as CACO [8], M-CACO [9], CIAC [10], ACA [21], we have performed some tests with the same test functions than those used in the related articles. Results are presented in Table 4.3.The value given in square bracket describes a test with a fixed number of evaluations, without any other stopping criterion and empty cells stand when values are not given in the literature [8,22]. Due to some lack of information in the literature, the comparison cannot be exhaustive, notably the efficiency and some pieces of information are lacking for CACO and M-CACO. Table 4.3 shows that APSA works better than other continuous ACO algorithm in most tests. It can be noticed that the average CPU time is sometimes a little higher than those of ACA. (See Table 4.4.)

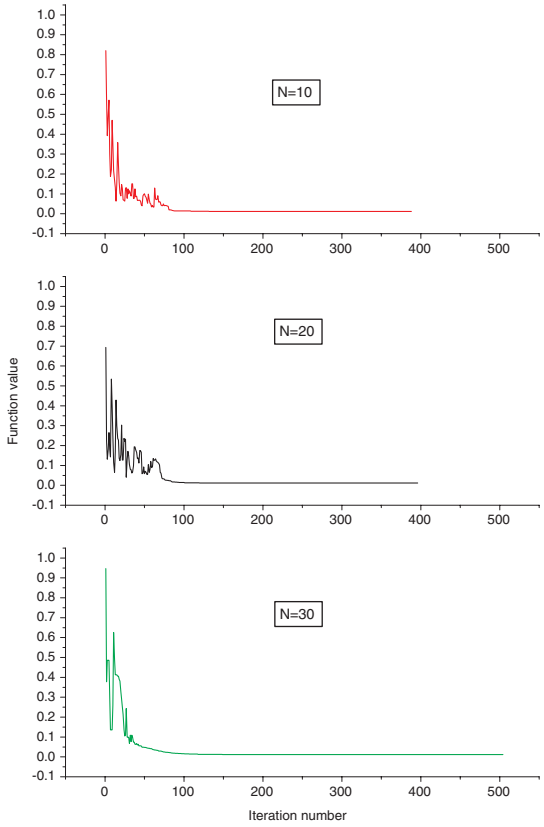

Fig. 4.2. Comparison results of APSA with different N.

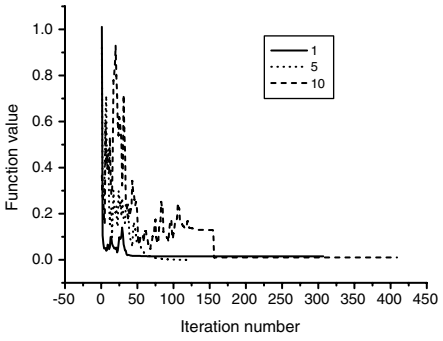

Fig. 4.3. Comparison results of APSA with different Δ_0.

Table 4.2
Performance of APSA with different dimension

Fun.	Dim.	Interval	Err	Per	Time (s)	Evals
Rosenbrock	5	[−5.21 5.21]	9.6575e−5	100	132.59	725
	10		9.8945e−5	100	266.45	894
	50		6.3698e−5	100	6598.9	4436
Griewangk	5	[−5.21 5.21]	5.6952e−5	100	23.24	182
	10		8.5642e−5	100	34.45	256
	15		6.3322e−5	100	75.54	122
	50		5.422e−5	100	1334.23	268

Table 4.3
Comparison of APSA performance with other continuous ACO

Fun.	APSA				CAIC			CACO		
	%	Err	Time (s)	Evals	%	Err	Evals	%	Err	Evals
R2	100	9.3443e−5	56.33	563	100	3e−3	11797	100	0.00	6842
SM6	100	3.1422e−5	3.45	15	100	9e−10	50000	100		22050
Gr5	100	5.6734e−5	3.76	14	63	0.01	48402			
Gr10	100	9.257e−5	53.24	15	52	0.05	50121	100	0.00	50000
GP	100	4.4435e−5	2.22	38	56	1.51	23391	100		5330
MG	100	8.323e−5	1.45	32	20	0.34	11751	100		1688
Bf1	100	6.573e−6	186.22	58	0	99521	50000		544	[200000]

Table 4.4
Comparison of APSA performance with other continuous ACO

Fun.	M-CACO		ACA			
	%	Evals	%	Err	Time (s)	Evals
R2	100	6842	100	2e−4	12.387	4698
SM6	100	22,050	100	4e−8	16.568	8963
Gr5			65	0.006	56.214	26,843
Gr10	100	50,000	100	0.015	156.23	46,217
GP	100	5330	100	0.560	25.397	32,574
MG	100	1688	100	1.984	43.158	28,156
Bf1			5	65,241	3125.5	50,000

Continues with Table 4.3.

Table 4.5
Comparison of APSA performance with other typical algorithms

Fun.	APSA			GA			DE		TS		
	%	Err	Evals	%	Err	Evals	Per	Evals	%	Err	Evals
R2	100	9.3070e−5	621	100	0.004	960	100	615	100	0.02	480
R5	100	9.8518e−5	719	100	0.15	3990			100	0.08	2142
SM6	100	7.1712e−5	14	100	0.0002	750	100	392	100	3e−8	338
Gr10	100	5.9357e−5	15	100		200000	100	12804			
GP	100	4.2824e−5	85	100	0.001	410			100	2e−3	231
MG	100	9.3587e−5	215	100		2844					

Finally, we compared the average solution quality obtained on problem instances taken from [10], with APSA to several literature results, such as the genetic algorithm (GA) [23], tabu search (TS) [24], differential evolution (DE) [25]. The comparison results shown as Table 4.5. One can notice, APSA is most comparable in term of accuracy.

5. Conclusion

The main contribution of the work is the development of a class of the continuous ant colony optimization algorithm. The convergence theory that guarantees the algorithms convergence almost surely to a stationary point for any continuous differentiable function and the simulation results with the benchmark typical functions shown the effectiveness of APSAs.

As future scope of work, one important issue consists of extending the algorithms by adapting them to the various classes of problems, such as nonlinearly optimization problems, multi-objective optimization problems and mixed variable (discrete and continuous space) optimization problems.

Appendix

The test functions used in the paper are described according to the following notations:

Dim. The dimension of the function.
$\vec{x}$. The variables vector.
Glob.Min. or Glob.Max. The value and (if known) the position of the global optimum.

I. Gr_n (Griewangk): $\mathrm{Gr}_n(\vec{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$.
Dim: n.
Glob.Min : $\mathrm{Gr}_n(0, 0, \ldots, 0) = 0$.

II. R_n (Rosenbrok): $\mathrm{R}_n(\vec{x}) = \sum_{i=1}^{n} [100(x_i^2 - x_{i+1}^2) + (1 - x_i)^2]$.
Dim: n.
Glob.Min: $\mathrm{R}_n(1, 1, \ldots, 1) = 0$.

III. SM_n (Sphere model): $\mathrm{SM}_n(\vec{x} = \sum_{i=1}^{n} x_i^2)$
Dim: n.
Glob.Min: $\mathrm{SM}_n(0, 0, \ldots, 0) = 0$.

IV. GP (Goldstein Price): $\mathrm{GP}(\vec{x}) = (1 + (x_1 + x_2 + l)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))(30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$.
Dim: 2.
Glob.Min: $\mathrm{GP}(0, -1) = 3$.

V. MG (Martin Gaddy): $\mathrm{MG}(\vec{x}) = (x_1 - x_2)^2 + ((x_1 + x_2 - 10)/3)^2$.
Dim: 2.
Glob.Min: $\mathrm{MG}(5, 5) = 0$.

VI. Bf_1 (Baluja f_1): $Bf_1(\vec{x}) = (\varepsilon + \|y_1| + \sum_{i=1}^{1} 00|y_i\|^{-1})$ with $y_1 = x_1$, $y_i = x_i + y_i - 1$
Dim: 100.
Glob.Max: $B_{f-1}(0, 0, \ldots, 0) = 10,000$.

References

[1] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1997) 53–66.
[2] Aaron C. Zecchin, Angus R. Simpson, Holger R. Maier, John B. Nixon, Parametric study for an ant algorithm applied to water distribution system optimization, IEEE Transactions on Evolutionary Computation 9 (2005) 175–191.
[3] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization with ILP preprocessing in multi-objective project portfolio selection, European Journal of Operational Research 171 (2006) 830–841.
[4] Marc Reimann, Marco Laumanns, Savings based ant colony optimization for the capacitated minimum spanning tree problem, Computers and Operations Research 33 (2006) 1794–1822.
[5] Feng Yuanjing, Feng Zuren, Ant colony system hybridized with simulated annealing for flow-shop scheduling problems, WSEAS Transaction on Business and Economics 1 (2004) 133–138.
[6] Sung-shun Weng, Yuan-Hung Liu, Mining time series data for segmentation by using ant colony optimization, European Journal of Operational Research 173 (2006) 921–937.
[7] Feng Yuanjing, Feng Zuren, Peng Qinke, An intelligent hybrid optimization strategy and its application, Journal of Xi'an Jiaotong University 138 (2004) 14–18.
[8] G. bllchev, I.C. Parmee, The ant colony metaphor for searching continuous design spaces, Lecture Notes in Computer Science 993 (1995) 25–39.
[9] Mohit Mathur, Sachin B. Karale, Sidhartha Priye, Ant colony approach to continuous function optimization, Industrial and Engineering Chemistry Research 39 (2000) 3814–3822.
[10] J. Dreo, P. Slarry, Continuous interacting ant colony algorithm based on dense heterarchy, Future Generation Computer Systems 20 (2004) 841–856.
[11] M. Blrattari, G. Di caro, M. Dorigo, For a formal foundation of the ant programming approach to combinatorial optimization. Part 1: The problem, the representation, and the general solution strategy, ATR Human Information Processing Laboratories, Kyoto, Japan, Tech Rep.TR-H-301, December 2000.
[12] N. Meuleau, M. Dorigo, Ant colony optimization and stochastic gradient descent, Artificial Life 8 (2002) 103–121.
[13] Walter J. Gutjahr, A graph-based ant system and its convergence, Future Generation Computer Systems 16 (2000) 873–888.

[14] Thomas Stutzle, Marco Dorigo, A short convergence proof for a class of ant colony optimization algorithm, IEEE Transactions on Evolutionary Computation 6 (2002) 358–365.
[15] T. Stutzle, H. Hoos, Max–min ant system, Future Generation Computer Systems 16 (2000) 889–914.
[16] Feng Yuanjing, Yu. Ll, Feng Zuren, A metropolis criterion based ant system and it's convergence analysis, International Conference on Sensing, Computing, and Automation (2006) 1633–1637.
[17] Virginia Torczon, On the convergence of pattern search algorithms, SIAM Journal of Optimization 7 (1997) 1–25.
[18] W.E. Hart, A convergence analysis of unconstrained and bound constrained evolutionary pattern search, Evolutionary Computation 9 (2001) 1–23.
[19] W.E. Hart, Evolutionary pattern search algorithms for unconstrained and linearly constrained optimization, IEEE Transactions on Evolutionary Computation 5 (2001) 388–397.
[20] L. Wang, X.P. Wang, Q.D. Wu, Ant system algorithm based Rousenbrock function optimization in multi-dimension space, in: Proceedings of First International Conference on Machine Learning and Cybernetics, Beijing, 2002, pp. 710–714.
[21] Feng Yuanjing, Feng Zuren, An immunity-based ant system for continuous space multi-modal function optimization, in: Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26–29 August 2004, pp. 1050–1054.
[22] N. Monmarch, G. Venturini, M. Slimane, On how *pachycondyla apicalis* ants suggest a new search algorithm, Future Generation Computer Systems 16 (2000) 937–946.
[23] R. Chelouah, P. Siarry, A continuous genetic algorithm designed for the global optimization, Journal of Heuristics 6 (2000) 191–213.
[24] R. Chelouah, P. Siarry, Tabu search applied to global optimization, European Journal of Operational Research 123 (2000) 256–270.
[25] R. Storn, K. Price, Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR95-012, International Computer Science Institute, Berkeley, CA.