Expert Systems with Applications 37 (2010) 4805–4810

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Ant colony optimization algorithm with mutation mechanism and its applications

Nan Zhao *, Zhilu Wu, Yaqin Zhao, Taifan Quan

School of Electronics and Information Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

ARTICLE INFO

Keywords:
Artificial intelligence
Ant colony optimization
Mutated evolutionary algorithm
Traveling salesman problem
CDMA
Multiuser detection

ABSTRACT

Mutated ant colony optimization (MACO) algorithm is proposed by introducing the mutation mechanism to the ACO algorithm, and is applied to the traveling salesman problem (TSP) and multiuser detection in this paper. Ant colony optimization (ACO) algorithms have already successfully been used in combinatorial optimization, however, as the pheromone accumulates, we may not get a global optimum because it can get stuck in a local minimum resulting in a bad steady state. The presented MACO algorithm can enlarge searching range and avoid local minima by randomly changing one or more elements of the local best solution, which is the mutation operation in genetic algorithm. As the mutation operation is simple to implement, the performance of MACO is superior with almost the same computational complexity. MACO is applied to TSP and multiuser detection, and via computer simulations it is shown that MACO has much better performance in solving these two problems than ACO algorithms.

1. Introduction

Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behaviors of insects and of other animals. In particular, ants have inspired a number of methods and techniques among which the most studied and the most successful one is the general purpose optimization technique known as ant colony optimization (ACO). ACO algorithms take inspiration from the foraging behavior of some ant species (Goss, Aron, Deneubourg, & Pasteels, 1989). These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. ACO exploits a similar mechanism for solving optimization problems (Dorigo, Birattari, & Stützle, 2006; Sim & Sun, 2003).

The first ACO algorithm, ant system (AS), is proposed as a means of solving the traveling salesman problem (TSP) (Dorigo, Maniezzo, & Colorni, 1996). AS has gained a great success in solving combinatorial optimization problems, however, the performance of it is still worse than some other metaheuristic algorithms (Laguna & Glover, 1993; Zhen-Ping & Bavarian, 1992). So, many other ACO algorithms are proposed inspired by AS, the performance of which is improved remarkably (Dorigo et al., 2006). The main ACO algorithms presented in the literatures are: ant-Q (Dorigo & Gambardella, 1996), ant colony system (ACS) (Dorigo & Gambardella, 1997), MAX–MIN ant system (MMAS) (Stützle & Hoos, 2000), rank-based ant system (Bullnheimer, Hartl, & Strauss, 1999), ANTS (Maniezzo, 1999), hyper-cube ant system (Blum, Roli, & Dorigo, 2001), KCC-Ants (Naimi & Taherinejad, 2009), and PDACO (Wu, Zhao, Ren, & Quan, 2009). The development of bio-inspired methodologies based on ant colony inspired algorithm systems is an emergent research area with applications in areas such as robotics (Lerman, Galstyan, Matinolli, & Ijspeert, 2002), quadratic assignment problems (Colorni, Dorigo, & Maniezzo, 1991), TSP (Li & Gong, 2003), and feature subset selection (Sivagaminathan & Ramakrishnan, 2007).

The most successful ACO algorithms are MMAS and ACS. Though MMAS and ACS can overcome the drawbacks of AS and achieve much better performance, it can get stuck in a local minimum resulting in a bad steady state as the pheromone accumulates. Genetic Algorithm (GA) is a powerful tool to solve combinatorial optimizing problems, and many new ACO algorithms have been proposed by introducing GA to the traditional ACO algorithms and achieve much better performance (Kaveh & Shahrouzi, 2008; Lee, Su, Chuang, & Liu, 2008). However, the GA-based ACO algorithms are extremely complicated and more time consuming. So in this paper, only the simple mutation operation of GA is introduced to the ACO algorithm, and mutated ant colony optimization (MACO) algorithm is proposed, which can improve the global searching capability with almost the same computational complexity as the ACO algorithm. The presented MACO algorithm can enlarge searching range and avoid local minima by randomly changing one or more elements of the local best solution, which is the mutation operation in genetic algorithm. So the performance of MACO is much better than the corresponding ACO algorithms with almost the same computational complexity. For

* Corresponding author. Address: Harbin Institute of Technology, 342#, Harbin, Heilongjiang 150001, China. Tel./fax: +86 0451 86418284.

E-mail addresses: zhaonan@hit.edu.cn (N. Zhao), wuzhilu@hit.edu.cn (Z. Wu), yaqinzhao@hit.edu.cn (Y. Zhao), quantf@sina.com (T. Quan).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.12.035

MACO will not change the working function of ACO, it can be combined with many ACO algorithms to improve the performance of them. So, mutated MMAS (M-MMAS) and mutated ACS (M-ACS) are proposed by combining MACO with MMAS and ACS respectively, and then applied to TSP. Through simulations it is shown that M-MMAS and M-ACS have better performance in solving TSP than their corresponding ACO algorithms.

Code Division Multiple Access (CDMA) has been the subject of extensive research in the field of mobile radio communications. This technique permits a large number of users to communicate simultaneously on the same frequency band; however, it also creates multiple-access interference (MAI). The MAI makes the conventional detector (CD), which can demodulate only one spread-spectrum signal without considering other signals, unreliable and insensitive to near-far effect in a multiuser environment. For this reason multiuser detection, which can overcome this problem, is a hot topic now for CDMA systems (Moshavi, 1996). The optimal multiuser detector (OMD) (Verdu, 1986) proposed by Verdu, is shown to be near-far resistant and has the optimal performance, however, the exponential complexity in the number of users makes it impractical to use in current CDMA systems. Therefore, research efforts have been concentrated on the development of suboptimal detectors, which exhibit good near-far effect resistant properties, have low computational complexity and achieve relatively high performance, such as MMSE detector (Xie, Short, & Rushforth, 1990), Hopfield neural network detector (Kechriotis & Manolakos, 1996), and stochastic cellular neural network detector (Wu, Zhao, Zhao, & Ren, 2007).

ACO can also be used in multiuser detection as a kind of suboptimal detectors, in which the length of the tour in TSP is related to the objective function of the OMD (Hijazi & Natarajan, 2004). In this paper, MACO multiuser detector is proposed by applying MACO algorithm to multiuser detection. Via simulations, it is shown that the MACO multiuser detector has a much better performance in reducing the near-far effect than the ACO multiuser detector and PDACO multiuser detector, which was proposed in our previous paper (Wu et al., 2009), as well as a superior performance in bit-error rate (BER).

The remainder of this paper is organized in three sections. In Section 2, some preliminaries about ACO are reviewed, and MACO is presented with its key characteristics. In Section 3, MACO is combined with ACS and MMAS, and applied to TSP. Simulation results are also presented. In Section 4, MACO is applied to multiuser detection. The performance of the MACO multiuser detector is compared with the ACO multiuser detector, PDACO multiuser detector, and some other detectors.

2. Ant colony optimization algorithms

Many ACO algorithms have been proposed. Here we present the original AS, its two most successful variants: MMAS and ACS, and the proposed MACO algorithm. In order to illustrate the differences between these algorithms, we use the TSP as a concrete example.

2.1. Ant system

AS is the first proposed ACO algorithm. Its main characteristic is that, after each iteration, the pheromone values are updated by all the M ants that have built solutions. The pheromone τ_{ij}, associated with the edge joining cities i and j, is updated as follows:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{m=1}^{M}\Delta\tau_{ij}^{m}, \tag{1}$$

where ρ is the evaporation rate, M is the number of ants, and $\Delta\tau_{ij}^{m}$ is the quantity of pheromone laid on edge (i,j) by ant m:

$$\Delta\tau_{ij}^{m} = \begin{cases} Q/L_m & \text{if ant } m \text{ used edge } (i,j) \text{ in its tour}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where Q is a constant, and L_m is the length of the tour constructed by ant m.

In the construction of a solution, ants select the following city to be visited through a stochastic mechanism. When ant m is in city i and has so far constructed the partial solution s^p, the probability of going to city j is given by:

$$p_{ij}^{m} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{c_{il}\in N(s^p)}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}} & \text{if } c_{ij}\in N(s^p), \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where $N(s^p)$ is the set of feasible components; that is, edges (i,l) where l is a city not yet visited by the ant m. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij}, which is given by:

$$\eta_{ij} = \frac{1}{d_{ij}}, \tag{4}$$

where d_{ij} is the distance between cities i and j.

AS has gain a great success in solving TSP, however, as the scale of TSP increases the performance of AS decreases seriously compared with other metaheuristic algorithms. So, most of the research on ACO has been focused on the methods to improve AS. The most successful ones are ACS and MMAS.

2.2. MAX–MIN ant system

MMAS is an improvement on the original AS. Its characterizing elements are that only the best ant updates the pheromone trails and that the value of the pheromone is bounded. The pheromone update is implemented as follows:

$$\tau_{ij} \leftarrow \left[(1-\rho)\cdot\tau_{ij} + \Delta\tau_{ij}^{\text{best}}\right]_{\tau_{\min}}^{\tau_{\max}}, \tag{5}$$

where $\tau_{\max}$ and $\tau_{\min}$ are respectively the upper and lower bounds imposed on the pheromone; the operator $[x]_b^a$ is defined as:

$$[x]_b^a = \begin{cases} a & \text{if } x > a, \\ b & \text{if } x < b, \\ x & \text{otherwise}, \end{cases} \tag{6}$$

and $\Delta\tau_{ij}^{best}$ is:

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} 1/L_{\text{best}} & \text{if } (i,j) \text{ belongs to the best tour}, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where L_{best} is the length of the tour of the best ant. This may be either the best tour found in the current iteration (iteration-best, L_{ib}) or the best solution found since the start of the algorithm (best-so-far, L_{bs}) or a combination of both.

2.3. Ant colony system

The most interesting contributions of ACS are the introduction of a local pheromone update in addition to the pheromone update performed at the end of the construction process (called offline pheromone update).

The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} \leftarrow (1-\varphi)\cdot\tau_{ij} + \varphi\tau_0, \tag{8}$$

where $\phi\in(0,1]$ is the pheromone decay coefficient, and τ_0 is the initial value of the pheromone.

The main goal of the local update is to diversify the search performed by subsequent ants during one iteration: by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and, hence, to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration.

The offline pheromone update, similarly to MMAS, is applied at the end of each iteration by only one ant, which can be either the iteration-best or the best-so-far. However, the update formula is slightly different:

$$\tau_{ij} = \begin{cases} (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij} & \text{if } (i,j) \text{ belongs to the best tour,} \\ \tau_{ij} & \text{otherwise,} \end{cases} \tag{9}$$

as in MMAS, $\Delta\tau_{ij} = 1/L_{\text{best}}$, where L_{best} can be either L_{ib} or L_{bs}.

Another important difference between ACS and AS is in the decision rule used by the ants during the construction process. In ACS, the so-called pseudorandom proportional rule is used: the probability for an ant to move from city i to city j depends on a random variable q uniformly distributed over [0,1], and a parameter q_0; if $q \leqslant q_0$, then $j = \arg\max_{c_{il}\in N(s^p)}\{\tau_{il}\eta_{il}^{\beta}\}$, otherwise Eq. (3) is used.

2.4. Mutated ant colony optimization

Genetic Algorithm (GA) is a powerful tool to solve combinatorial optimizing problems and it is first proposed by John Holland professor in 1975 (Holland, 1992). It solves the formulated optimization problem by using the idea of Darwinian evolution. Basic evolution operations, including crossover, mutation and selection, make GA be apt to perform global search very effectively. In order to solve the problems which are not very complex and whose real-time requirement is strict, such as multiuser detection, only the mutation mechanism is introduced to the ACO algorithm, and a MACO algorithm is proposed. The presented MACO algorithm can enlarge searching range and avoid local minima by randomly changing one or more elements of the local best solution after each iteration.

In the iteration of MACO algorithm, assuming that the local best solution after the nth iteration is $s_{nbest} = (s_{1nbest}, \ldots, s_{Knbest})^T$. Randomly choose one or more elements in the s_{nbest}, change them in a certain manner, and keep the other elements unchanged. Through this mutation operation, the mutated solution s'_{nbest} of s_{nbest} can be got. If s'_{nbest} is better than s_{nbest}, replace s_{nbest} by s'_{nbest}. Otherwise the local best solution remained unchanged. The number of the mutated elements is decided by the number K of all the elements in the solution of the problem, and the larger K is, the more elements can be mutated. If K is extremely large, the number of mutated elements can be set to decrease as iterations carried on.

Through introducing the mutation mechanism to ACO algorithm, the local searching performance is enhanced, the diversity of solutions is expanded, and early convergence can be avoided.

3. MACO algorithm for TSP

Since MACO can be combined with many ACO algorithms to improve the performance of them, M-MMAS and M-ACS are proposed by combing MACO with MMAS and ACS respectively, and applied to TSP.

3.1. M-MMAS and its application to TSP

M-MMAS is proposed by combining MACO with MMAS, and it follows the principles of MMAS described in Section 2.2 except the mutation operation after each iteration.

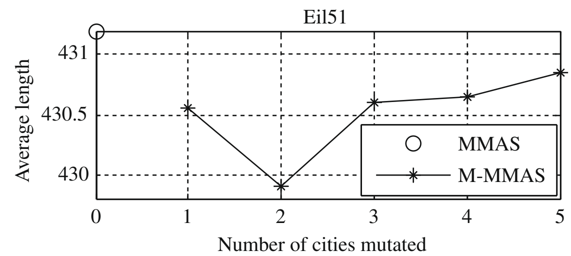

Fig. 1. Average length vs. number of cities mutated of M-MMAS in Eil51.

To compare M-MMAS with MMAS, they are applied to two TSP problems, respectively (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/). The first is Eil51, and there are 51 cities in it with the best solution 425. Both MMAS and M-MMAS run for 1000 iterations using 10 ants, and especially in M-MMAS several cases with different number of mutated elements (cities), which is increased for 1–5, are considered. The randomly chosen cities are mutated by exchanging their positions with their next cities. Through simulations, the average length of the tour in Eil51 versus the number of mutated cities in each case of M-MMAS curves (the number of mutated cities of MMAS is zero) are depicted in Fig. 1.

In Fig. 1, it is shown that the average lengths of Eil51 solved by M-MMAS in all cases with different number of cities mutated are all shorter than that solved by MMAS.

The second TSP problem is St70, and there are 70 cities in it with the best solution 675. Both MMAS and M-MMAS run for 1500 iterations using 10 ants, and especially in M-MMAS several cases with different number of mutated elements (cities), which is increased for 1–7, are considered. The randomly chosen cities are mutated by exchanging their positions with their next cities. Through simulations, the average length of the tour in St70 versus the number of mutated cities in each case of M-MMAS curves (the number of mutated cities of MMAS is zero) are depicted in Fig. 2.

In Fig. 2, it is shown that the average lengths of St70 solved by M-MMAS in all cases with different number of cities mutated are all shorter than that solved by MMAS.

Through solving these two typical TSP problems, Eil51 and St70, using MMAS and M-MMAS respectively, we can see that the performance of M-MMAS in solving combinatorial optimization problems is superior to that of MMAS.

3.2. M-ACS and its application to TSP

M-ACS is proposed by combining MACO with ACS, and it follows the principles of ACS described in Section 2.3 except the mutation operation after each iteration.

To compare these two algorithms, ACS and several cases of M-ACS with different number of mutated cities are applied to the Eil51 and St70 TSP problems, respectively. The parameters are

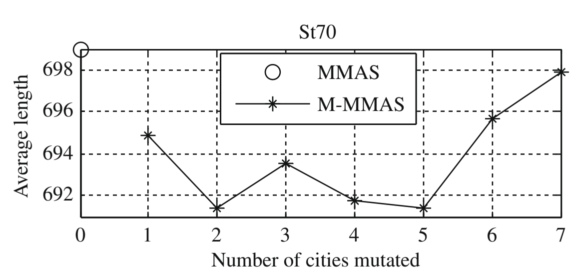

Fig. 2. Average length vs. number of cities mutated of M-MMAS in St70.

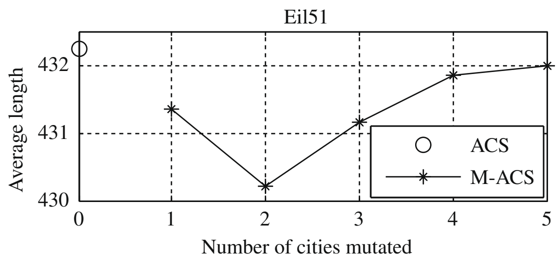

Fig. 3. Average length vs. number of cities mutated of M-ACS in Eil51.

the same as those described in 3.1. Through simulations, the average lengths of the tour in Eil51 and St70 versus the number of mutated cities in each case of M-ACS curves (the number of mutated cities of ACS is zero) are depicted in Figs. 3 and 4.

In Figs. 3 and 4, it is shown that the average lengths of Eil51 or St70 solved by M-ACS in all cases with the different number of cities mutated are all shorter than that solved by ACS. So, we can see that the performance of M-ACS in solving combinatorial optimization problems is superior to that of ACS.

For MACO only introduces mutation mechanism at the end of each iteration and will not change the working function of ACO, it can be combined with many ACO algorithms to improve the performance of them. In Sections 3.1 and 3.2, M-MMAS and M-ACS are proposed by combining MACO with MMAS and ACS respectively, and then they are used in solving TSP. Simulation results show that by randomly changing one or more elements of the local best solution after each iteration, the presented MACO algorithm can enlarge searching range and avoid local minima, and the performance of ACO algorithms improved by MACO in solving combinatorial optimization problems is superior to that of the corresponding algorithm with the same computational complexity. We can also see that the number of cities mutated should not be too large, and if so, the better solution of the local best solution can not be obtained because of the great change of it. The proper number of elements mutated should be decided by the scale of the problem to be solved.

4. MACO multiuser detector

4.1. Multiuser detection

CDMA has been the subject of extensive research in the field of mobile radio communications. This technique permits a large number of users to communicate simultaneously on the same frequency band. However, this creates MAI, which makes the CD of demodulating a spread-spectrum signal in a multiuser environment unreliable and insensitive to near-far effect. For this reason multiuser detection, which can overcome this problem, is a hot topic now for CDMA systems.

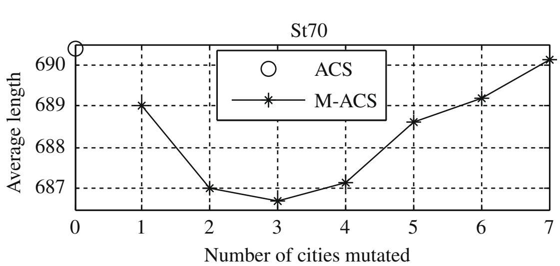

Fig. 4. Average length vs. number of cities mutated of M-ACS in St70.

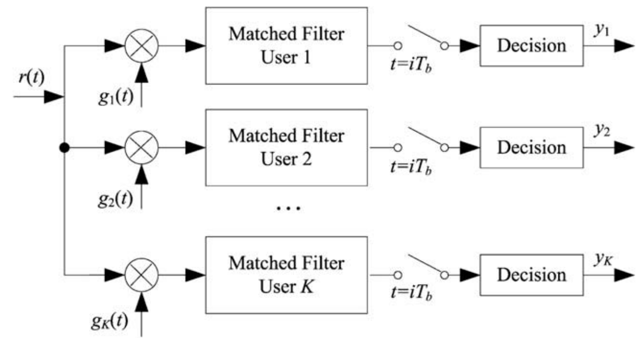

Fig. 5. The conventional detector.

4.1.1. Conventional detector

Assuming there are K users of a CDMA system in a synchronous single-path channel, the received signal can be expressed as:

$$r(t) = \sum_{k=1}^{K} A_k(t) g_k(t) d_k(t) + n(t), \tag{10}$$

where $A_k(t)$, $g_k(t)$, and $d_k(t)$ are the amplitude, signature code waveform, and information of the kth user, respectively. $n(t)$ is additive white Gaussian noise (AWGN), with a two-sided power spectral density of $N_0/2$ W/Hz.

The CD is composed of a bank of K matched filters, and can be shown in Fig. 5.

In Fig. 5, the existence of MAI has a significant impact on the capacity and performance of the CD system because the CD follows a single-user detector strategy. As the number of interfering users increases, the amount of MAI increases.

4.1.2. Optimal multiuser detector

Verdu has shown that the OMD may be achieved by producing an estimate for the information vector transmitted based on the maximization of the logarithm of the likelihood function. The objective function of the OMD is given as:

$$b_{opt} = \arg\max\{2Y^T Ab - b^T Hb\}, \tag{11}$$

where $b \in \{+1, -1\}$, $Y^T = (y_1, \ldots, y_K)$ is the row vector consisting of the sampled outputs of the matched filters,A is the diagonal matrix consisting of the corresponding received amplitudes, and $H = A^T RA$, in which R is a $K \times K$ uniform correlation matrix.

Despite the huge performance and capacity gains over the CD, the OMD is not practical. The exponential complexity in the number of users makes the cost of this detector too high. Consequently, research efforts have been concentrated on the development of suboptimal multiuser detectors that exhibit good near-far resistance, reasonable implementation complexity and comparable BER performance to that of the OMD.

4.2. ACO multiuser detector

As the speciality of multiuser detection in the CDMA system, the ACO algorithms should be adjusted if we want to apply them to the multiuser detection. The adjustments are as follows.

(1) For the K users in the systems are independent, without losing generality, we can let each ant travel in the fixed order from the 1st user to the Kth user. In this case, the ants should not decide whether the user has been traveled.
(2) As there is not any heuristic information, we can discard parameters η_{ij}, α and β. Because the transmitted information by each user can only be +1 or -1, the probability of what value the kth user transmitted decided by the ant m at the time t is given by:

$$p_{kj}^{m}(t) = \frac{\tau_{kj}(t)}{\sum_{s\in(-1,+1)}\tau_{ks}(t)}, \quad k = 1,2,\ldots,K, \; j = +1 \text{ or } -1. \tag{12}$$

(3) The decision of which is the best solution is based on the values of the objective function in Eq. (14), and the solution that has the largest value is the best.
(4) Because the datum in multiuser detection should be processed in real-time, only the best ant in the current iteration deposits the pheromone after each iteration, but the pheromone on all the paths still evaporates.

Through the rules set above, multiuser detection can be described as a path-choosing problem which can be solved by ACO algorithms.

4.3. MACO multiuser detector

ACO algorithms have been applied to multiuser detection successfully; however, the performance of the ACO multiuser detector still can be improved. MACO has been proved to have better performance than ACO in Sections 3.1 and 3.2, so MACO can be used in this field and MACO multiuser detector is proposed.

The mutation mechanism used in the MACO multiuser detector can be described as follows. Assuming that the local best solution after the nth iteration is $b_{nbest} = (b_{1nbest},\ldots,,b_{Knbest})^T$. Randomly choose one or more bits in the b_{nbest}, handle them with logical not operator, and keep the other bits unchanged. Through this mutation operation, the mutated solution b'_{nbest} of b_{nbest} can be got. If b'_{nbest} is better than b_{nbest}, replace b_{nbest} by b'_{nbest}. Otherwise the local best solution remained unchanged. The number of the mutated bits is decided by the number of users in the system K, and the larger K is, the more bits are mutated. If K is extremely large, the number of mutated bits can be set to decrease as iterations carried on.

MACO is applied to multiuser detection following the rules described in Section 4.2, and the MACO multiuser detector can be got. It is represented by the following steps:

Step 1 Initialize of parameters, including the number of iterations N_c, the population of the ant colony M, the evaporation rate ρ, and the initial value of the pheromone $\tau(0)$.
Step 2 Set the outputs of the matched filters in Fig. 5 as the initial global best solution.
Step 3 M ants travel from the 1st user to the Kth user following Eq. (12), and then we can get M solutions in the nth iteration.
Step 4 Compare the M solutions based on the Eq. (11), and set the solution that has the largest value (equal to C_n) of Eq. (11) as the local best solution b_{nbest} in this iteration.
Step 5 Calculate the mutated solution b'_{nbest} of b_{nbest} following the mutation mechanism described above and get the value of Eq. (11) C'_n using s'_{nbest}. Compare C_n and C'_n. If C'_n is larger than C_n, replace b_{nbest} by b'_{nbest} and replace C_n by C'_n.
Step 6 Update the pheromone as follows:

$$\tau_{kj}(t+1) = (1-\rho)\cdot\tau_{kj}(t) + \Delta\tau_{kj}, \quad k = 1,2,\ldots K, \quad j = +1 \text{ or } -1, \tag{13}$$

$$\Delta\tau_{kj} = \begin{cases} (C_n + Q)/r & (k,j) \in b_{nbest}, \\ 0 & \text{otherwise}, \end{cases} \tag{14}$$

where $\Delta\tau_{kj}$ is the quantity of pheromone laid on edge (k,j) of the kth user with the value j,Q is a positive constant to ensure $(C_n + Q) \geqslant 0$, and r is a constant to adjust the value of $\Delta\tau_{kj}$.
Step 7 Compare the local best solution with the global best solution. If the local best solution is better than the global best solution, set the local best solution as the global best solution.

Table 1
Computational complexity comparison.

Detector	Calculation number
CD	$2KN$
ACO multiuser detector	$[2K(K+N)+5K+1]MN_c$
MACO multiuser detector	$[2K(K+N)+5K+1]MN_c$
OMD	$[2K(K+N)+5K+1]2^K$

Step 8 Output the global best solution if stopping criterion is satisfied, or return to Step 3.

The computational complexity of an algorithm can be measured by the number of multiplications and additions. The computational complexity CD, ACO multiuser detector, MACO multiuser detector and OMD is compared in Table 1 when there are K users in the system and each detects only one bit data.

In Table 1, N is the length of PN sequences, M is the number of the ants in the colony, N_c is the number of iterations. In this paper, they are set as $K = 10$, $M = 10$, and $N_c = 20$, so the computational complexity of ACO multiuser detector and MACO multiuser detector is the same and 80.5% lower relative to OMD, and it will be much less than that of OMD when K is even larger.

4.4. Experimental results

In order to evaluate the performance of the MACO multiuser detector, a DS-CDMA system using it is designed as Fig. 6.

In Fig. 6, the number of users $K = 10$ in the CDMA system and the length of PN sequences used is 15. The number of the ants $M = K = 10$, the number of iterations $N_c = 20$, the evaporation rate $\rho = 0.3$, the parameters Q and r in Eq. (14) are equal to 200 and 15, and there is only one bit of information transmitted mutated in each iteration. The received signal $r(t)$ is handled in the matched filter bank, the outputs of which are fed into the MACO multiuser detector, and then we can get the estimate of the baseband information transmitted of each user. A variety of simulation experiments are presented comparing the performance of the CD, the ACO multiuser detector, the PDACO multiuser detector (Wu et al., 2009), the MACO multiuser detector, and the OMD in the system depicted in Fig. 6.

First, the near-far effect resistant performance of these detectors is compared. In order to illustrate explicitly, only the transmitted energy of the first user E_1 changes, and the energy of other users E_k $(k = 2,3,\ldots,K)$ is all unchanged with their signal-noise ratios (SNR) = 6 dB.

From the simulation results in Fig. 7, we can see that near-far effect resistant performance of the MACO multiuser detector is much better than the CD, the ACO multiuser detector and the PDACO multiuser detector, and is close to OMD, especially when the near-far effect is serious.

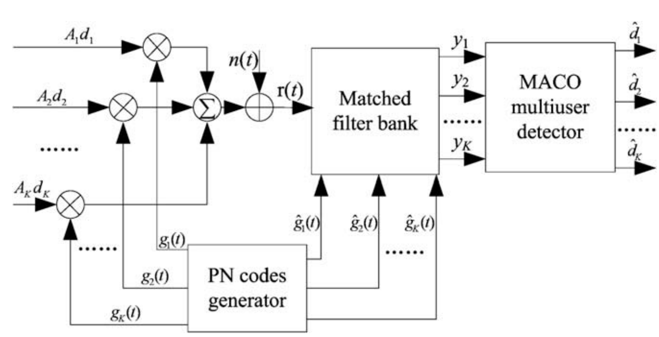

Fig. 6. The block diagram of a MACO multiuser detection system in DS-CDMA.

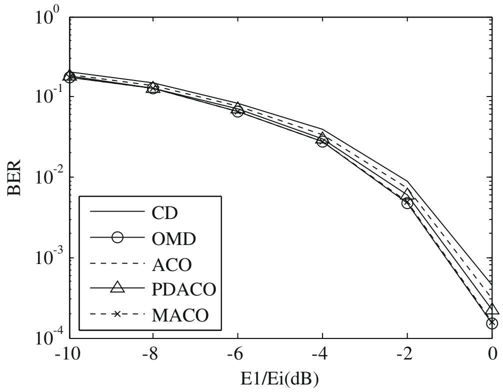

Fig. 7. Bit-error rate vs. near-far ratio.

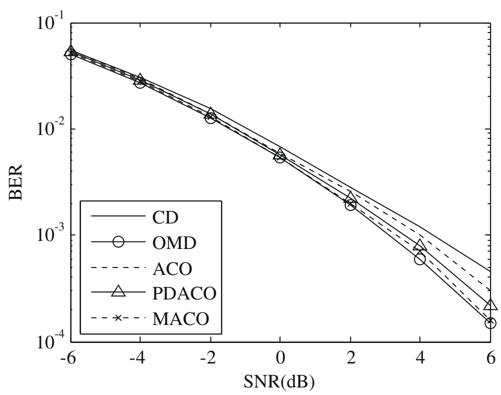

Fig. 8. Bit-error rate vs. signal-noise ratio.

Second, the performance of these detectors with no near-far effect is compared. In this case, the transmitted energy of all the users is ensured to be equal ($E_1 = E_k$, $k = 2,3,\ldots,K$) and unchanged.

It is shown in Fig. 8 that if the near-far effect is not considered, the BER of the MACO multiuser detector is much lower than that of the CD, the ACO multiuser detector and the PDACO multiuser detector, and is close to the BER of OMD.

Therefore, the simulation results show that the overall performance of the MACO multiuser detector is much better than the ACO multiuser detector and the PDACO multiuser detector, and is close to or even equal to OMD by introducing the mutation mechanism to the ACO algorithm, so it is more suitable as a suboptimal multiuser detection scheme in CDMA systems.

5. Conclusion

In this paper we have proposed a MACO algorithm, which can enlarge searching range and avoid local minima by introducing the mutation mechanism to the ACO algorithms. So, the performance MACO is superior to that of ACO with almost the same computational complexity. The applications of MACO are discussed and it is used in TSP and multiuser detection. Since MACO can be combined with many ACO algorithms to improve the performance of them, M-MMAS and M-ACS are proposed by combining MACO with MMAS and ACS respectively, and applied to TSP. Simulating results show that M-MMAS and MACS have much better performance than the corresponding ACO algorithms with the same computational complexity. MACO multiuser detector is proposed and computer simulations show that the performance of both the BER and near-far resistant is superior to that of ACO multiuser detector.

References

Blum, C., Roli, A., & Dorigo, M. (2001). HC-ACO: The hyper-cube framework for ant colony optimization. In *Proceedings of MIC'2001-metaheuristics international conference*, Porto (pp. 399–403).

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A new rank-based version of the ant system: A computational study. *Central European Journal for Operations Research and Economics, 7*(1), 25–38.

Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. In *Proceedings of the european conference on artificial life*, Paris, France (pp. 134–142).

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine, 1*(4), 28–39.

Dorigo, M., & Gambardella, L. M. (1996). A study of some properties of Ant-Q. In *Fourth international conference on parallel problem solving from nature*, Berlin (pp. 656–665).

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53–66.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 26*(1), 29–41.

Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften, 76*(12), 579–581.

Hijazi, S. L., & Natarajan, B. (2004). Novel low-complexity DS-CDMA multiuser detector based on ant colony optimization. In *IEEE 60th vehicular technology conference* (pp. 1939–1943).

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge, MA: MIT Press<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.

Kaveh, A., & Shahrouzi, M. (2008). Optimal structural design family by genetic search and ant colony approach. *Engineering Computations, 25*(3), 268–288.

Kechriotis, G. I., & Manolakos, E. S. (1996). Hopfield neural network implementation of the optimal CDMA multiuser detector. *IEEE Transactions on Neural Networks, 7*(1), 131–141.

Laguna, M., & Glover, F. (1993). Integrating target analysis and tabu search for improved scheduling systems. *Expert Systems with Applications, 6*(3), 287–297.

Lee, Z. L., Su, S. F., Chuang, C. C., & Liu, K. H. (2008). Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. *Applied Soft Computing Journal, 8*(1), 55–78.

Lerman, K., Galstyan, A., Matinolli, M., & Ijspeert, A. (2002). A macroscopic analytical model of collaboration in distributed robotic systems. *Artificial Life, 7*(4), 375–393.

Li, Y., & Gong, S. H. (2003). Dynamic ant colony optimization for TSP. *International Journal of Advanced Manufacturing Technology, 22*(7–8), 528–533.

Maniezzo, V. (1999). Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORMS Journal on Computing, 11*(4), 358–369.

Moshavi, S. (1996). Multi-user detection for DS-CDMA communications. *IEEE Communications Magazine, 34*(10), 124–136.

Naimi, H. M., & Taherinejad, N. (2009). New robust and efficient ant colony algorithms: Using new interpretation of local updating process. *Expert Systems with Applications, 36*(1), 481–488.

Sim, K. M., & Sun, W. H. (2003). Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Transactions on Systems, Man and Cybernetics, Part A, 33*(5), 560–572.

Sivagaminathan, R. K., & Ramakrishnan, S. (2007). A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert Systems with Applications, 33*(1), 49–60.

Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. *Future Generation Computer Systems, 16*(8), 889–914.

Verdu, S. (1986). Minimum probability of error for asynchronous Gaussian multiple-access channels. *IEEE Transactions on Information Theory, 32*(1), 85–96.

Wu, Z. L., Zhao, N., Ren, G. H., & Quan, T. F. (2009). Population declining ant colony optimization algorithm and its applications. *Expert Systems with Applications, 36*(3P2), 6276–6281.

Wu, Z. L., Zhao, N., Zhao, Y. Q., & Ren, G. H. (2007). Stochastic cellular neural network for CDMA multiuser detection. In *Fourth international symposium on neural networks*, Nanjing, China (pp. 651–656).

Xie, Z., Short, R. T., & Rushforth, C. K. (1990). A family of suboptimum detectors for coherent multiuser communications. *IEEE Journal on Selected Areas in Communications, 8*(4), 683–690.

Zhen-Ping, L., & Bavarian, B. (1992). Optimization of job scheduling on parallel machines by simulated annealing algorithms. *Expert Systems with Applications, 4*(3), 323–328.